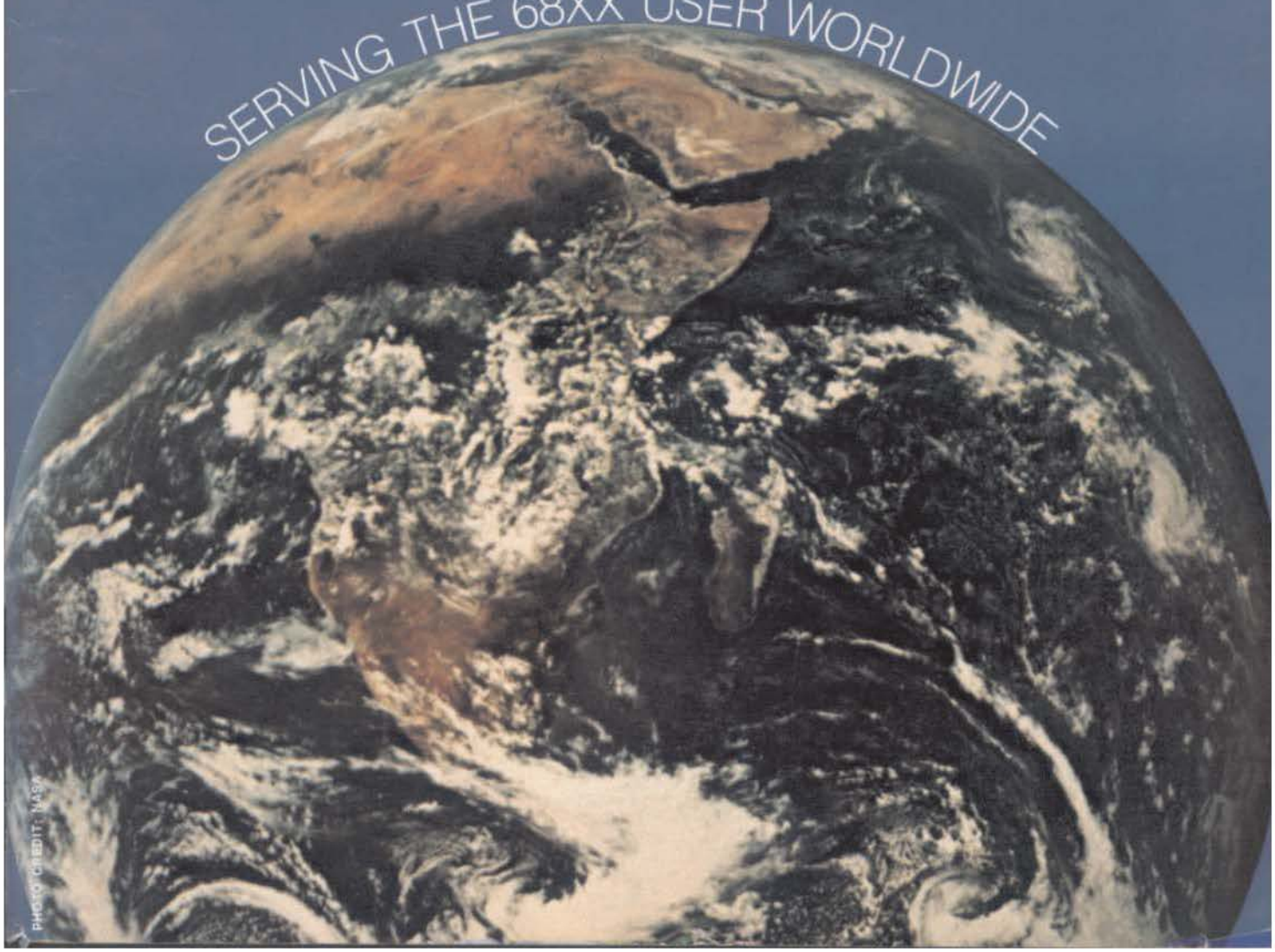
'68' MICRO JOURNAL

$2.95 USA

Australia	A $ 4.75.	New Zealand	NZ $ 4.75
Singapore	S $ 9.45	Hong Kong	H $23.50
Malaysia	M $ 9.45	Sweden	30.-SEK

VOLUME IV ISSUE VIII • Devoted to the 68XX User • August 1982

"Small Computers Doing Big Things"



YOUR CHOICE-smart either way

- Over 140 software driven functions
- 82 x 24 or 82 x 20 screen format — software selectable
- High resolution 7 x 12 matrix characters — P-31 green phosphor
- Upper/lower case character set — plus graphics character set
- 56-key alphanumeric keyboard — plus 12-key cursor, numeric pad
- Internal editing functions — insert, delete, scroll, roll, slide, etc.
- Parallel printer I/O port
- 50 to 38,400 baud operation — programmable
- Cursor type, cursor position, print control characters, protected fields, shift inversion, dual intensity and many other features

8212 — twelve-inch diagnonal screen or **8209** — nine-inch diagnonal screen

SWTPC

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

Pascal for 6809

Pascal for the 6809 is a true native code compiler. Unlike the usual P-code Pascals which run in an interpretive manner, ours produces efficient assembly language mnemonics which can be assembled and run directly. This compiler is available for both 6809 FLEX™ and UniFLEX™. Many features not found in other Pascal systems were implemented while avoiding those features completely non-standard. Features of the Pascal system include:

- Supports most of Jensen and Wirth specification
- Produces fast and efficient 6809, native code
- FLEX run-time package may be trimmed
- Double precision real numbers (16.8 digits)
- Implements scalar, subrange and structured data types
- Standard I/0 using file buffer pointers
- Dynamic storage allocation
- Ability to call other Pascal programs
- FLEX version may call assembly language programs
- Buffered or single character terminal input
- Standard math functions: SIN, COS, ARCTAN, EXP, LN, SQR, SQRT
- Random number generator function
- Many usable, sample programs included
- UniFLEX version supports:
  - Random file positioning
  - Ability to call various UniFLEX system routines
  - Ability to execute UniFLEX utility commands

Pascal on diskette for 5" and 8" 6809 FLEX is available for $200.00 The 5" version requires two disk drives. The UniFLEX version is $300.00 and includes one year of maintenance. All orders should include 3 percent for postage and handling (10 percent on foreign orders).

™FLEX and UniFLEX are trademarks of Technical Systems Consultants, Inc.

technical systems consultants, inc.

111 Providence Rd., Chapel Hill, N.C. 27514

(919) 493-1451

'68' MICRO JOURNAL

Portions of text prepared using the following.

SWTPC 6800-6809-DMAF2-CDS1-CT82-Sprint 3
Southwest Technical Products
219 W. Rhapsody
San Antonio, Texas 78216

EDITOR - WORD PROCESSOR
Technical Systems Consultants, Inc.
Box 2573, W. Lafayette, IN 47906
FLEX is TM of TSC

GIMIX Super Mainframe-Assorted memory boards
GIMIX Inc.
1337 West 37th Place
Chicago, Il 60609

TABLE OF CONTENTS

Vol. IV, Issue VIII — August '82

Send All Correspondence To:

Computer Publishing Center
68 MICRO JOURNAL
5900 Cassandra Smith
PO Box 849
Hixson, TN 37343
615 842-4600

68 Micro Journal is published 12 times a year by Computer Publishing, Inc. Second class postage paid at Hixson, Tennessee and additional entries. Postmaster: send Form 3579 to 68 Micro Journal, PO Box 849, Hixson, Tennessee.

SUBSCRIPTION RATES
USA
1-Year $24.50 2-Years $42.50 3-Years $64.50
FOREIGN
See Page 52

Items Submitted for Publication

Articles submitted for publication should be accompanied by the authors full name, address, date and telephone number. It is preferred that articles be submitted on either 5 or 8 inch diskette in TSC Editor format or STYLO format. All diskettes will be returned.

The following TSC Text Processor commands ONLY should be used (due to our proportional processor): .sp space, .pp paragraph, .fi fill and .nf no fill. Also please do not format within the text with multiple spaces. The rest we will enter at time of editing.

STYLO commands are all acceptable except the .pg page command, we print edited text files in continous text.

All articles submitted on diskettes should be in TSC FLEX™ format, either FLEX2 6800, or FLEX9 6809 any version.

If articles are submitted on paper they should be on white 8X11 bond or better grade paper. No hand written articles (hand written or drawn art accepted). All paper submitted articles will be photo reproduced. This requires that they be typed or produced with a dark ribbon (no blue), single spaced and type font no smaller than 'elite' or 12 pitch. Typed text should be approximately 7 inches wide (will be reduced to column width of 3 1/2 inches). **Please use a dark ribbon!**

All letters to the editor should also comply with the above and bear a signature. Letters of 'gripes' as well as 'praise' are solicited. We attempt to publish all letters to the editor verbatim, however, we reserve the right to reject any submission for lack of 'good taste'. We reserve the right to define what constitutes 'good taste'.

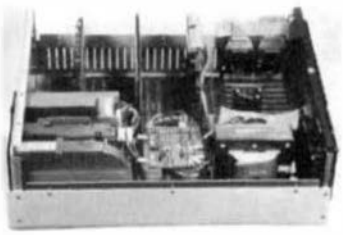
Advertising: Commercial advertisers please contact the 68 Micro Journal advertising department for current rate sheet and requirements.

Classified: All classified must be non-commercial. Maximum 20 words per classified ad. Those consisting of more than 20 words should be figured at .35 cents per word. 20 words or less $7.50 minimum, one time, paid in advance. No classified ads accepted by telephone.

FLEX - OS-9 LEVEL ONE - UNIFLEX - OS-9 LEVEL TWO

ONLY GIMIX Systems can be configured to run any of these.

GIMIX systems utilize the most powerful 6809 operating systems: FLEX, UniFLEX, OS-9 LEVEL ONE and TWO -- the systems the PROs use. This means a wide selection of software to choose from as well the ability to develop sophisticated, multi-user/multi-tasking programs on your **GIMIX** System.



The GIMIX CLASSY CHASSIS™ consists of a heavy-weight aluminum mainframe cabinet which provides more than ample protection for the electronics and 1 or 2 optional 5¼" drives.

Backpanel connectors can be added for convenient connection of terminals, printers, drives and other peripherals.

A 3 position locking keyswitch enables users to disable the front panel reset button to prevent accidental or unauthorized tampering with the system.

The **GIMIX** system mother board provides fifteen 50 pin slots and eight 30 pin I/O slots -- the most room for expansion of any SS50 system available. The on board baud rate generator features 11 standard baud rates, 75 to 38.4K, for maximum versatility and compatibility with other systems. Extended address decoding allows the I/O block to be addressed anywhere in the 1 megabyte address space. All components feature Gold plated connectors for a lifetime of solid connections. All boards are fully buffered for maximum system expansion.

Each **GIMIX** Mainframe System is equipped with an industrial quality power supply featuring a **ferro-resonant constant voltage transformer** to insure against problems caused by adverse power input conditions such as A.C. line voltage fluctuations etc. The supply provides 8 volts at 30 amps and plus or minus 16 volts at 5 amps, more than enough capacity to power a fully loaded system and two internal drives.

The 2MHz GIMIX 6809 PLUS CPU board includes a time of day clock with battery back-up and 6840 programmable timer to provide the programmer with convenient, accurate time reference. Later addition of 9511 or 9512 arithmetic processors is provided for on the board. The unique **GIMIX** design enables software selection of either OS-9 or FLEX, both included in many complete **GIMIX** systems.

GIMIX STATIC RAM boards require no complicated refresh timing cycles or clocks for data retention. GIMIX memory boards are guaranteed for 2 MHz operation with no wait state or clock stretching required.

Our low power NMOS RAM requires less than 3/4 amp at 8V for a fully populated 64K board. For critical situations, our non-volatile 64K byte CMOS static RAM boards with built in battery back-up retain data even with system power removed. A fully charged battery will power this board for a minimum of 21 days. A write protect switch permits CMOS boards to be used for PROM/ROM emulation and software debugging.

The GIMIX DMA controller leaves the processor free to perform other tasks during disk transfers - an important feature for multi-user/multi-tasking systems where processor time allocation is critical. The DMA board will accomodate up to 4 drives 5¼" or 8" in any combination running single or double density single or double headed. Programmed I/O Disk Controllers are also available.

GIMIX systems are designed with ultimate **RELIABILITY** in mind. You can choose from the below featured systems or select from our wide variety of components to build a custom package to suit your needs.

GIMIX 2MHz 6809 System including: CLASSY CHASSIS, 6809 PLUS CPU BOARD, 56KB STATIC RAM, 2 SERIAL PORTS W/CABLES, GMXBUG MONITOR, FLEX, and OS-9 LEVEL 1 ..... **$3248.49**
FOR TWO 5¼" 40 TRACK DSDD DRIVES ADD ..... **$ 900.00**
GIMIX 128KB WINCHESTER SYSTEM including: CLASSY CHASSIS, 6809 PLUS CPU BOARD, 128KB STATIC RAM, 4 SERIAL PORTS W/CABLES, 5¼" 80 TRACK DSDD FLOPPY DISK DRIVE, 19MB 5¼" WINCHESTER HARD DISK, OS9 LEVEL 2, EDITOR AND ASSEMBLER **$8998.09**

50HZ Versions Available, 8" Drives Available — Contact GIMIX for Prices and Information.

The Sun Never Sets On A GIMIX!

GIMIX users are found on every continent, including Antarctica. A representative group of GIMIX users includes: **Government Research and Scientific Organizations** in Australia, Canada, U.K. and in the U.S.; NASA, Oak Ridge, White Plains, Fermilab, Argonne, Scripps, Sloan Kettering, Los Alamos National Labs, AURA. **Universities:** Carleton, Waterloo, Royal Military College, in Canada; Trier in Germany; and in the U.S.; Stanford, SUNY, Harvard, UCSD, Mississippi, Georgia Tech. **Industrial users** in Hong Kong, Malaysia, South Africa, Germany, Sweden, and in the U.S.; GTE, Becton Dickinson, American Hoechst, Monsanto, Allied, Honeywell, Perkin Elmer, Johnson Controls, Associated Press, Aydin, Newkirk Electric, Revere Sugar, HI-G/AMS Controls, Chevron. **Computer mainframe and peripheral manufacturers,** IBM, OKI, Computer Peripherals Inc., Qume, Floating Point Systems. **Software houses;** Microware, T.S.C., Lucidata, Norpak, Talbot, Stylo Systems, AAA, HHH, Frank Hogg Labs, Epstein Associates, Softwest, Dynasoft, Research Resources U.K., Microworks, Meta Lab, Computerized Business Systems.

GIMIX Inc. reserves the right to change pricing and product specifications at any time without further notice.

GIMIX® and GHOST® are registered trademarks of GIMIX Inc.
FLEX and UniFLEX are trademarks of Technical Systems Consultants Inc.
OS-9 is a trademark of Microware Inc.

1337 WEST 37th PLACE
CHICAGO, ILLINOIS 60609
(312) 927-5510
TWX 910-221-4055

GIMIX INC.

The Company that delivers

Quality Electronic products since 1975.

Build performance into your system

with OS-9** software tools

Unix*-based, multitasking, modular, and versatile: these key features are some of the reasons why more 6809 computer manufacturers have selected OS-9 as their standard operating system than any other. And OS-9 has been put to work by thousands of users in almost every conceivable computer application in business, science, industry, education, and government.

Your operating system should not be a barrier between you and your computer. OS-9 is very friendly and easy to use. Its modular structure makes it easy to customize, plus its comprehensive documentation shows you exactly how to interface it to just about any I/O device.

OS-9's advanced features unleash the performance potential of almost any 6809 computer — large or small. In many respects the OS-9/6809 combination is more powerful than many *minicomputers!*

There are two basic versions of OS-9. Both have the same basic features and capabilities. OS-9 Level One runs on small to medium sized systems having up to 64K memory. The Level Two version runs on medium to large size systems having memory management hardware and up to 1 megabyte of memory, and includes record and file locking for multiuser database applications.

Here are just a few reasons why you should insist on OS-9 for your microcomputer system.

- Over 40 utility commands
- Friendly "Shell" command interpreter
- Tree-structured multilevel file directories
- Full timesharing support with log-in and file security
- Fast, secure random and sequential access files
- Comprehensive English language error messages
- Compact real-time multitasking executive
- Hardware or software memory management
- Device independent interrupt-driven I/O
- Fully ROMable for small control systems
- Standard versions available from manufacturers of most popular 6809 computers

OS-9 PASCAL Language Compiler

- most complete and versatile PASCAL available for the 6809
- capable of generating P-code for interpretive execution while debugging OR
- highly optimized 6809 assembly language source code output for maximum speed
- "virtual memory" P-code interpreter lets you run large PASCAL programs

CIS COBOL * Compiler**

- ideal for most demanding business applications
- features ISAM, Debug, ACCEPT/DISPLAY and Interprogram Communications modules
- retains full compatibility with CP/M software
- meets ANSI 1974 Level One COBOL standard and is GS certified

Also available-FORMS 2 automatic program generator for easy interative design of screen oriented applications.

BASIC09 Structured Basic Interactive Compiler**

- fastest and most comprehensive full Basic language available for the 6809
- combines standard Basic with the best features of PASC L
- features compiler speed, interpreter friendliness and superlative debugging facilities
- option available: Run B...a ROMable run-time system for compiled Basic 09

C Language Compiler

- complete implementation of the UNIX version 7 C language
- includes INT, CHAR, SIGNED, UNSIGNED, FLOAT AND LONG data types, structures, unions, standard C library and a full preprocessor with macro definitions
- generates fully reentrant 6809 assembly language source code output

For more information contact your computer supplier or

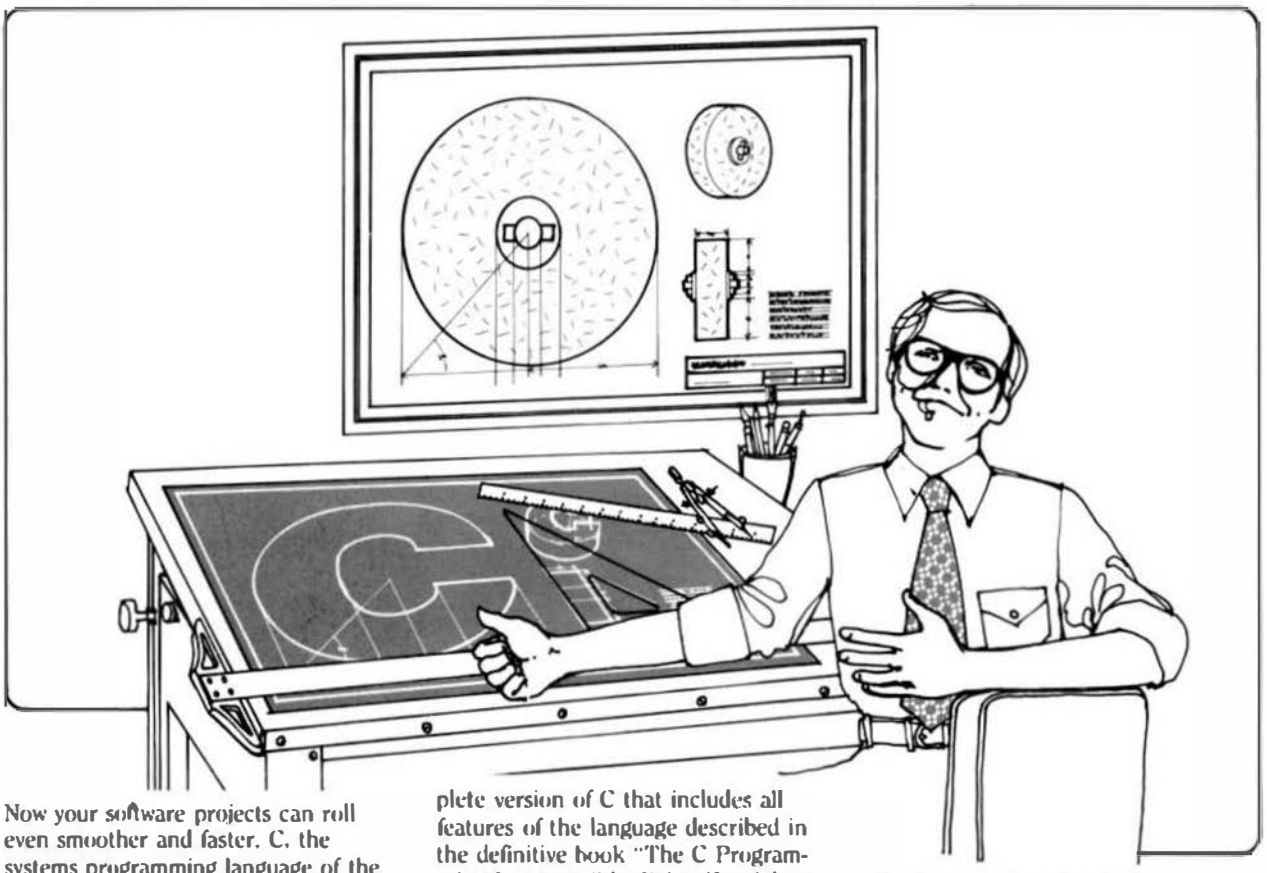
MICROWARE

Microware Systems Corporation
5835 Grand Avenue, Des Moines, Iowa 50312 515-279-8844 • Telex 910-520-2535

*Unix is a trademark of Bell Laboratories. ***CIS Cobol is a trademark of Micro Focus, Inc. OS-9** and Basic09 are trademarks of Microware and Motorola, Inc.

C:

The Greatest Invention Since The Wheel



Now your software projects can roll even smoother and faster. C, the systems programming language of the future, is here today for Microware's OS-9 Operating System. Professionals rave about C because it's a structured language that can handle the most demanding real-time applications or painlessly produce simple system software.

When performance counts . . .

Few languages can match C's outstanding ability to produce fast, compact native code. In fact, it is one of a very few languages that is truly efficient enough to be used to produce operating systems, critical real-time programs, and compilers. Because of the richness and variety of C operators and the way they naturally combine, complex functions require less code. Plus the 6809 architecture makes it a superior C machine.

Complete and standard . . .

Microware's new C compiler is a complete version of C that includes all features of the language described in the definitive book "The C Programming Language" by Brian Kernighan and Dennis Ritchie. OS-9 C features: preprocessor with conditional compilation; complete standard function library; char, int, long, and float data types; pointers, register variables, arrays, structures, and unions; one-pass compilation; and assembly language source code output.

The bridge to Unix and the future . . .

Because Microware's C compiler has essentially all features of Unix C, and because the OS-9 operating system is a Unix-type operating system, C programs readily move between OS-9 and Unix. And it is becoming apparent that C will be the preferred programming language for all popular 16-bit microcomputers. As a result, software written in C is inherently protected against processor obsolescence and is assured portability to all latest-generation microprocessors including the 68000.

Plus the OS-9 connection . . .

C is the latest member of the broadest line of 6809 software tools in the industry: Microware's OS-9 family. All OS-9 system functions are directly callable from C programs. The C compiler utilizes the standard OS-9 Text Editor and Assembler, and can process data files used by Basic09, Pascal, and Cobol.

Write or call for our free catalog. We accept phone orders and MasterCard and VISA orders.

Unix is a trademark of Bell Telephone Laboratories. OS-9 is a trademark of Microware Systems Corp.

MICROWARE.

Microware Systems Corporation
5835 Grand Ave., Des Moines, IA 50312
(515) 279-8844 Telex 910-520-2535

DynaStar

WORD PROCESSING SYSTEM FOR OS-9

OS-9 USERS:

If your computer has a SCREEN and you're still struggling with an editor that only knows about LINES, then obviously YOU don't know about

DynaStar

DynaStar is a powerful, menu-driven screed editor equally suited to the tasks of program preparation and document processing. With the addition of the optional DynaForm print formatter, it is the best word-processing package you can buy for your OS-9 system.

DynaStar Version II is now available and features no-nonsense "what you see is what you get" editing for virtually any terminal with or without cursor addressing (it must be at least able to go to "home"). To edit, simply place the cursor where you want it, and type. Any printable character you type is entered directly into your text, and any non-printable control character causes immediate execution of an editing command. Single keystroke commands permit movement of the cursor in any direction, by character, tab, word, line, or screen full, and deletion of characters, words (left or right) or a whole line. Two keystroke commands augment this set by moving the cursor to the left margin, top or bottom of the screen, beginning or end of the edit buffer, or the beginning of the next paragraph. You can search for any string, replace with any other, do it again, mark original blocks of text, copy, move or delete blocks, read or write to side-files, set tabs and margins, or center the current line.

DynaStar features automatic word-wrap, and it can right-justify text as you enter it so you will see exactly how it will look *before* you print it. If you later make alterations or change the margins, you can reform the text a paragraph at a time with two keystrokes. For programmers, there is a special automatic indent mode to help you write well-structured code. DynaStar includes a Shell command which lets you do almost anything (including edit another file) without even losing your place in your current document, and it permits editing of large disk files in stages without forcing you to break up your files.

If you want to define more powerful commands, DynaStar includes a macro facility which lets you convert any control character to one or a string of characters of your choice. You can use this feature to create global search-and-replace commands, insert "boiler-plate," or simply re-map your keyboard. You can also provide a special "start-up string" which is automatically executed whenever you enter the editor to set up modes such as auto-justify, display a directory, define your favorite macros, or re-map the keyboard.

For compiste word-processing, we offer our DynaForm text formatter which provides all the standard features such as pagination, headers and footers with page numbers, single space, double space, multiple space, **bold face**, **double-strike**, and underline. DynaForm has its own macro facility with string variables, nested include files, a full merge-print capability for generating form letters and mailing lists, and it can generate an index automatically, sorted alphabetically or by page number. You can call it from DynaStar to proof-print the active edit buffer, or by itself to print a disk file while you edit another.

DynaStar II (for the faithful) $149.95
DynaForm text formatter $149.95
Both purchased together: $275.00
Note: DynaStar Version I (no macros) will be available at the original price until May 31, and current owners may upgrade to version II with full credit until June 30.

AVAILABLE SOON FOR FLEX 9

Spelltest

From Dale Puckett

FOR OS-9 AND FLEX

SPELLTEST is the most versatile 68XX spelling checker available.
MENU'S MAKE OPERATION EASY. From the menu you may: Print a list of suspect words; Print a list of valid words; Check each suspect word one by one; Read your text, stopping to check suspect words; Use additional dictionaries for more thorough checking or special applications; Build an additional dictionary of newly accepted words; Write correct text file to disk.
While checking you may: Accept the suspect word. Accept and save in the dictionary. Replace with correct spelling.
Designed to be used by the layman, SPELLTEST is right at home in the office. Ease of use and speed will recover the cost in days.
22,000 word dictionary covers the first 25,000 entries in the American Heritage listing of the most common English words.
500 built in common words (and, or, the, etc.) and 300 specific to your field; filter the text and allows a large file to processed even in small computers.

PRICE $199.00

A/BASIC

Basic Compiler

For OS-9 and FLEX

If you are still programming in assembler, this is the program for you! This BASIC compiler generates pure, fast efficient 6809 machine code from easy to write BASIC source programs. Uses ultra-fast integer math, extended string functions, boolean operators and run-time operations. Output is ROMable and RUNS WITHOUT ANY RUN-TIME PACKAGE. Supports IF-THEN-ELSE structure, random access, and several improvements over the original 6800 version sold by Microware. Optimized for the 6809, A/BASIC is 8 to 10 times faster than the original 6800 version, and produces code approximately 30% smaller.

SPECIAL

CHESS program coded in A/BASIC (originally sold for $50) is included FREE on the disk in both source and object for your enjoyment. Also some utilities are included for testing and examples, all in source on the disk!

ONLY $150.00
specify OS-9 or FLEX

PLOT

Now you can have GRAPHICS added to all your programs. Just write the data out to a virtual array and call PLOT. PLOT is written in TSC XBASIC and the source is included on the disk.
INFINITE RESOLUTION GRAPHICS ON YOUR TERMINAL OR PRINTER. HISTOGRAMS, BARGRAPHS, XY PLOTS PLUS OTHERS.
IN TSC XBASIC SOURCE INCLUDED ON DISK. $44.95

TOOLKIT NO1

The Basic Programmers *Toolkit*
by Dick Bartholomew

The Basic Programmers Toolkit gives the BASIC programmer the power and flexibility never before achieved under FLEX.

PRICE $49.95 object only
$69.95 with source on disk!

TOOLKIT NO2

The Programmers *Toolkit*
by Dick Bartholomew

The Programmers Toolkit is a package of utilities and programs that extend the capabilities of FLEX to the utmost.

PRICE $49.95 object only
$69.95 with source on disk!

Dynasoft PASCAL 1.4 for OS-9

Dynasoft Pascal 1.4 includes all the features of the FLEX version 1.3 with the following enhancements: Chain, Fread, Fwrite, Seek, Open, Creat, Close, Delete, Fork, Send, Wait, Sleep, Setime, Time, Getstatus, Setstatus, SetPriority, GetProcID, and JSR. This is an excellent and fast program, small enough to write utilities but powerful enough for things like DynaStar.
Integer Only
Object only $69.95
Add for run-time source on disk $30.00
Add for source of Dynasoft Pascal itself $125.00

CRASMB

MULTI CPU CROSS ASSEMBLER FOR 6809 FLEX
by Frank Hoffman

CRASMB is a conditional macro assembler with the capability to use different CPU overlays in order to cross assemble. These CPU overlays called *'CPU PERSONALITY MODULES' (CPM's)* can be called from a source file, thereby making it easy to create object code for a variety of CPU's. It is also possible to create new CPM's yourself for any 8 or 16 bit CPU. The information needed is included in the manual. If you decide to do this, it would be advisable to purchase the source for one of the CPM's and modify it rather than starting from scratch. CPM's are currently available for the following CPU's: 6809, 6800, 6805, 6502, Z80, 8080, 1802, and others coming.

PRICE $139.95

Includes one 8 bit CPM of your choice (not source)
Additional CPM's
8 Bit $25.00 Source $25.00 extra

THE BILL PAYER SYSTEM™

THE BILL PAYER is a package of 10 menu driven programs in TSC Extended Basic. This powerful system helps you keep track of your bills. You can create a vendor list, enter invoices to be paid, generate reports about them, print your checks and much more. Uses random access files.

Explore Package now included at the same price.

THE PURCHASE ORDER system adds purchase orders to the BILL PAYER. This package of programs adds another level of control to your expenditures. Prints out purchase orders and keeps track of purchases. Requires the Bill Payer to work.

INCOME/EXPENSE LEDGER. This valuable package is most appreciated at tax time. Allows up to 99 income and expense numbers. Ties into the PURCHASE ORDER system, and the Bill Payer.

Includes manual and source supplied on disk in TSC Extended Basic.

THE BILL PAYER
PURCHASE ORDER
INC/EXP LEDGER
ALL FOR $169.95

COLOR COMPUTER USERS

THE POWERFUL FLEX DISK OPERATING SYSTEM WITH HUNDREDS OF SOFTWARE PACKAGES IS NOW AVAILABLE!

Now you can run FLEX, OS-9 and Radio Shack disk software on your Color Computer. If you have a 32K Color Computer with the Radio Shack disk system all you need to do is make a trivial modification to access the hidden 32K, as described in the Feb. issue of COLOR COMPUTER NEWS and the April issue of '68' Micro. You can get FLEX from us right now. OS-9 will be ready by summer. Please note that this will only work with the Radio Shack disk system and 32K/64K memory chips that RS calls 32K. Maybe they put 64K's in yours, too. If you don't have a copy of the article, send a legal size SASE (40¢ stamps) and we'll send it to you.

Using this system to run FLEX and OS-9 has many advantages. First, it gives you 48K from zero right up to FLEX. This means that ALL FLEX compatible software will run with *NO MODIFICATIONS* and *NO PATCHES*! There are no memory conflicts because we moved the screen up above FLEX which leaves the lowest 48K free for user programs.

What you end up with is 48K for user programs, 8K for FLEX and another 8K above FLEX for the screens and stuff. We have a multi screen format so you can page backward to see what scrolled by and a Hi Res screen that will enable us to have 24 lines by 42 character display is on the way. That's better than an Apple!

We also implemented a full function keyboard, with a control key and escape key. All ASCII codes can now be generated from the Color Computer keyboard!

We also added some bells and whistles to Radio Shack's Disk system when you're running FLEX or OS-9. We are supporting single or double sided, single or double density, 35, 40 and 80 track drives. If you use double sided drives, the maximum is three drives because we use the drive 3 select for side select. When you are running the Radio Shack disk, it will work with the double sided drives but it will only use one side and only 35 tracks. Using 80 track drives is okay, but will not be compatible with standard Radio Shack software. You can also set each drive's stepping rate and drive type. (SS or DS - SD or DD)

In case you don't understand how this works, I'll give you a brief explanation. The Color Computer was designed so that the roms in the system could be turned off under software control. In a normal Color Computer this would only make it go away. However, if you put a program in memory to do something first (like boot in FLEX or OS-9), when you turn off the roms, you will have a full 64K RAM System with which to run your program.

Now, we need the other half of the 64K ram chips to work, and this seems to be the case most of the time as the article states. Of course, you could also put 64K chips in.

Some neat utilities are included.

MOVEROM moves Color Basic from ROM to RAM. Because it's moved to RAM you can not only access it from FLEX, you can run it and even change it!! You can load Color Computer cassette software and save it to FLEX disk. Single Drive Copy, Format and Setup commands plus an online help system are included.

Installing FLEX is simple. Insert the disk and type:

RUN "FLEX"

That's all there is to it! You are now up and running in the most popular disk operating system for the 6809. There are hundreds of software packages now running under the FLEX system. Open your Color Computer to a whole new world of software with FLEX.

FLEX $89.00
INCLUDES OVER 25 UTILITIES!

Other languages available include; FORTH, Pascal, Fortran77, 'C', A/BASIC compiler, plus more.
Application packages include; A/R, G/L, A/P, Inventory, Electronic Spreadsheets, Accounting, Database programs and more. SEND FOR LIST.

TRS-80 COLOR COMPUTER COMPLETE WITH 64K RAM, 24K ROM, SINGLE DISK DRIVE AND FLEX, SET UP AND READY TO RUN FOR ONLY $1,275. Includes 120 day extended warranty. If you have a Computer, call about RS disk controllers and drives.

AUTOTASK

WITH MENU

AUTOTASK with MENU is a revolutionary new concept designed to overcome the problems and frustrations which confront the non-technical when using a computer. Users are greeted with a series of self-prompting interactive menus linking directly to the application. Several example menus are provided. You can create your own menus from simple text files. AUTOTASK with MENU gives you unlimited software flexibility by providing a system to coordinate multiple application programs.

Bundle any 7 different software packages to present a coordinated system to the user. AUTOTASK with MENU is compatible with all FLEX compatible software. It uses very little memory and is easy to learn.

PRICE $129.95

Includes source on disk!

Manual $10.00

6502 TRANSLATOR
Translator 6502 code to 6809
$75.00

INVENTORY
with MATERIAL
REQUISITION PLANNING
$100.00

SUPER SLEUTH
Disassembler for 6800/6809 or Z80
$99.00

TABULA RASA
Electronic Spreadsheet
$100.00

UNIFLEX SIMULATOR
Runs Under Flex
$100 Flex $110 UniFlex

TRS-80 COLOR COMPUTER

FORTH FOR THE TRS-80 COLOR COMPUTER DISK SYSTEM

Trying to get control of your Color Computer?? Tired of translating HEX to decimal?? Tired of remembering where the VDG and SAM are and how to program them?? Want to write machine language code with assembly language mnemonics instead of POKES?? Want to write programs in half the time?? Want to write lots of small pieces of code that you can put together in seconds to do BIG JOBS??? Want a language that is at least 5 to 10 times faster than BASIC??? Want to learn everything there is to know about FORTH, with the best manual on the market, including lots of examples of FORTH applications, and detailed explanations of how everything works??

CC FORTH IS THE ANSWER!!

Includes Editor, 6809 Assembler
String Functions, Disk Data File
Operations and Much More!

$99.95

FORTH

FLEX COMPATIBLE FORTH

BY Chuck Eaker, Ph.D.
X-FORTH NOTES
Supplied on one 8" disk or 2 5" disks, with a 400+ page manual.
Disk(s) have the source of everything but the core.
PRICE only $149.95 plus $2.50 S&H
Manual available seperately for $49.95 plus $2.50 S&H

We Have DynaCalc For Flex

$200.00

SOFTWARE CATALOG

PROGRAM	OBJECT/WITH CBASIC	SOURCE/CBASIC	code
BILLPAYER		169.95	x
LOT		44.95	x
TABULA RASA		100.00	x
Mailing List		99.95	x
Forms Display		49.95	x
Inventory with Material Requisition Planning		100.00	x
Some Common BASIC programs		69.95	x
Infomag Data Base Management System	295.00		x
Osborne Accounts Receivable	295.00		x
Osborne Accounts Payable	295.00		x
Osborne General Ledger	295.00		x
DynaCalc	200.00		9
UniFlex Simulator		110.00	9
FLEX For Color Computer	99.00		9
X-FORTH (FLEX)	149.95		8 & 9
CC-FORTH (TRS-80 Color)	99.95		9
TOOLKIT #1 (BASIC)	49.95	69.95	9
TOOLKIT #2	49.95	69.95	9
AUTOTASK		129.95	9
A/BASIC Compiler	150.00		9
Extended Utilities	49.95	69.95	9
Password Protection	69.95	89.95	9
CRASMBIX Assembler (1 INC)	139.95		9
Personality Modules (1 INC) 6502, 6800, 6805, 6809, Z80, 8080, 1802	25.00	50.00	9 ea
READTAPE		54.95	9
SELFTEST	199.00	299.00	9
READTEST	54.95	74.95	8 & 9
ESTHER	39.95	59.95	8 & 9
HELP	29.95	49.95	8 & 9
Job Control Program	49.95	89.95	8 & 9
DYNASOFT PASCAL (FLEX)	59.95	89.95	9
DYNASFT PASCAL (OS-9)	69.95	99.95	9
DYNASOFT Compiler Source		125.00	P
DYNASTAR Screen Editor (OS-9)	149.95		9
SUPER SLEUTH (6800/6809)		99.00	8 & 9
SUPER SLEUTH (Z80)		99.00	8 & 9
CROSS Assembler Macros for TSC ASMB 6800/1, 6805, 6502, Z80, 8080/5 3 FOR		49.95 99.95	each
6502 Translator		75.00	9
Debugging Simulators 6805 or 6502		75.00	9 ea
STYLOGRAPH 2.0	295.00		9
STYLOGRAPH MAIL MERGE	125.00		9
STYLOGRAPH Spelling Checker	145.00		9

CODE X = XBASIC, 9 = 6809, 8 = 6800, P = PASCAL

Software by Technical Systems Consultants, Inc.	
Flex™ (includes Editor & Assembler)	150.00
UniFLEX™ (includes one year maintenance and update)	450.00
Editor	50.00
Assembler	50.00
68000 Cross Assembler on 6809	250.00
Text Processor	75.00
Extended Basic	100.00
Basic Precompiler (specify standard or extended)	50.00
Pascal (Flex™)	200.00
Pascal (UniFLEX™) (Add $75.00 for one year's maintenance and update)	225.00
Soft/Merge Package	75.00
6809 Flex™ Utilities	75.00
Debug Package	75.00
Diagnostic Package	75.00
Software by Microware Systems Corp.	
OS-9™ Level One Operating System	200.00
OS-9™ Level Two Operating System	500.00
BASIC09™	200.00
OS-9™ Macro Text Editor	125.00
OS-9™ Interactive Assembler	125.00
OS-9™ Interactive Debugger (Disk version)	50.00
CIS Cobol Compiler	895.00
Pascal Compiler	400.00

USA add $2.50 for standard UPS shipping & handling
Foreign orders add 20% Airmail
Specify 5" or 8" disk and 6800 or 6809
VISA—MASTER CHARGE—DINERS CLUB ACCEPTED

OUR SOFTWARE IS GIMIX COMPATIBLE

FRANK HOGG LABORATORY
130 MIDTOWN PLAZA
SYRACUSE, NY 13210 (315) 474-7858

FLEX & RS COLOR COMPUTER

If you are tired of playing games on your TRS-80C™ Color Computer, or find that you are handicapped by the limitations of the RS BASIC in trying to write a Program that will allow you to actually USE the Color Computer as a COMPUTER; and if you have been studying the Advertisements in this Magazine and wishing that you could run THESE Programs on your Computer, YOU ARE READY TO MOVE UP TO THE FLEX9™ Operating System. If you want to have REAL PROGRAMMING POWER, using an Extremely Powerful Business BASIC, PASCALs, C Compilers, a full-blown Macro Assembler with a Library capability so you are not continuously "reinventing the wheel", YOU ARE READY TO MOVE UP TO THE FLEX9™ Operating System. If you would like to see if YOU REALLY COULD USE A COMPUTER IN YOUR BUSINESS, or begin to make your Computer start PAYING IT'S OWN WAY by doing some Computer Work for the millions of small businesses around you, such as Wordprocessing, Payroll, Accounting, Inventory, etc., then YOU ARE READY TO MOVE UP TO THE FLEX9™ Operating System. How?? DATA-COMP has the way!

DATA-COMP's FLEX9™ Conversion for the TRS-80C™ Color Computer was designed for the SERIOUS COMPUTER USER: with features like greatly increased Display Screens, WITH Lower Case Letters, so you can put a FULL Menu on ONE Screen, or see SEVERAL Paragraphs at the same time; with features like providing a FULL Keyboard so you have FULL Control of your Computer AND it's Programs NATURALLY, without needing a chart to see what Key Combination will give you what function; with USER ORIENTED functions to make using the Operating System natural, like having the Computer AUTOMATICALLY determine what type of Disk is being used in what type of Disk Drive and working accordingly, rather that you have to specify each and every thing for it, or like having the Computer work with the Printer you have been using all along without you having to tell the new Operating System what is there; etc., etc., etc.

DATA-COMP has everything you need to make your TRS-80C™ Color Computer WORK for YOU; from Parts and Pieces to Full, Ready To Use SYSTEMS. DATA-COMP designs, sells, services, and SUPPORTS Computer SYSTEMS, not just Software. CALL DATA-COMP TODAY to make your Computer WORK FOR YOU!

-- SYSTEM REQUIREMENTS --

FLEX9™ Special General Version w/Editor & Assembler (which normally sell for $50.00 ea.)	$150.00
F-MATE(RS) FLEX9 Conversion Rout. for the RS Disk Controller when purchased with Special General FLEX9 Sys.	$49.95
when purchased without the General FLEX9 Sys.	$59.95
NEW -- Full Source Code for the Conv. Routines	$159.95
Set of Eight 64K RAM Chips w/ Mod. instructions	$99.95
Color Computer with 64K RAM and EXT. BASIC	$649.95
Color Computer with 16K RAM	$375.95
C or Computer with 16K RAM and EXT. BASIC	$465.95

-- SPECIAL SYSTEM PACKAGES --

64K Radio Shack COLOR COMPUTER, Radio Shack COLOR DISK CONTROLLER, a Disk Drive System, Special General Version of FLEX9™, F-MATE(RS)™, and a Box of 10 Double Density Diskettes; a COMPLETE, ready to run SYSTEM on your Color TV Set.
$1379.95

-- FOR USERS THAT ALREADY HAVE FLEX9 & Disk Drives --

Radio Shack DISK CONTROLLER with F-MATE(RS)™ and a Special Two Drive DISK CABLE	$289.95

-- DISK DRIVE PACKAGES, with RS Controller --

These Packages Include the Radio Shack Disk Controller, Disk Drives with Power Supply and Cabinet, and Disk Drive Cable:

PAK #1 ==> 1 Single Sided, Double Density Sys.	$509.95
PAK #2 ==> 2 Single Sided, Double Density Sys.	$759.95
PAK #3 ==> 1 Double Sided, Double Density Sys.	$629.95
PAK #4 ==> 2 Double Sided, Double Density Sys.	$929.95

-- PARTS AND PIECES --

Radio Shack Disk Controller	$169.95
1 ea. Single Sided, Double Density Disk Drive	$249.95
1 ea. Double Sided, Double Density Disk Drive	$349.95
Single Drive Cabinet with Power Supply	$79.95
Double Drive Cabinet with Power Supply	$99.95
Single Drive Disk Cable for RS Controller	$24.95
Double Drive Disk Cable for RS Controller	$34.95
Micro Tech. Prods., Inc. LOWER CASE ROM Adapter	$74.95
Radio Shack BASIC Version 1.1 ROM	$34.95

SOFTWARE

CHESS 6809

Requires FLEX™ and one of the following CRT terminals
SWTPC CT-82 8212 8212-W

Features:

- Two display boards.
- Four levels of play.
- Point scoring system.
- Play white or black.
- Change or set-up boards/piece positions.
- Forfeit move.
- Swap sides.
- Make move and swap sides.
- Change skill level.
- Stop and restart game.
- Solve 'Mate in 1-2-3-4' moves.

$79.95 Specify 5" or 8" disk

This is one of the strongest CHESS programs running on any microcomputer, estimated uscf Rating 1600 +.

Note: Personal checks allow 3-4 week delivery.

DIET-TRAC Forecaster

A Diet Planning and Analysis Program

DIET-TRAC Forecaster is a program that plans a diet in terms of either calories and percentage of carbohydrates, proteins and fats (C/P/F %) or grams of Carbohydrate, Protein and Fat food exchanges of each of the six basic food groups (vegetable, bread, meat, skim milk, fruit and fat) for a specific individual.

Sex, Age, Height, Present Weight, Frame Size, Activity Level and Basal Metabolic Rate for normal individuals are taken into account. Ideal weight and sustaining calories for any weight of the above individual are calculated. When a weight goal is given (either gain or loss), and a calorie plan is agreed upon between the computer and the individual, the number of days to reach the weight goal is projected. The starting and ending rate of weight loss is calculated, and a daily calendar with each day's predicted weight for a 30-day period is printed.

FLEX VERSION	**$59.95**
UniFLEX VERSION	**$89.95**

PRINTERS

The Epson MX-80
$455.00
The Epson MX-100
$725.00

MX-70 $355.00 MX-80 FT $575.00

MEMORY

SWTPC-Motorola, MP32
32K Dynamic Memory Board
Assembled & Tested
1 MHZ - No extended addressing
Can be set up for $0-7FFF or 8000-FFFF

$149.95

MasterCard VISA

DATA-COMP SOUTH EAST MEDIA

P.O. Box 794 Chattanooga TN 37443
1-615-842-4601

Verbatim Diskettes.

5" Soft Sector Disks

Single Side Single Density	$2.75 ea.
Single Side Double Density	$2.75 ea.
Double Side Double Density	$4.92 ea.
Plastic Storage Box	$2.00 ea.

8" Soft Sector Disks

Single Side Single Density	$3.75 ea.
Single Side Double Density	$4.10 ea.
Double Side Double Density	$4.75 ea.
Plastic Library Box	$5.00 ea.

Foreign Orders Add 10% Surface—20% Air Mail

DRIVES & CABINETS W/PS

5 1/4"

TEAC - Single Sided, Double Density, 40 Track....,$249.95

TEAC - Double Sided, Double Density, 40 Track...,,$349.95

CABINET - Single Drive with Power Supply.....$ 79.95

CABINET - Double Drive with Power Supply.....$ 99.95

CABLE - Single Drive.....$ 24.95

CABLE - Double Drive.....$ 34.95

* NOTE - When ordering cables please specify S50 Bus or Other!!!

Call or write for disk controller Board information.

RUMORS & SUCH

Rumors, ramblings and Such

It has been my fancy, time to time, to unload on you my faithful and loyal readers, rumors, ramblings (about varied subjects) and my opinions on various subject-matter. Not always have the rumors I passed on been completely accurate, but for the most part they have been. Also I have received rebuttals to some of my ramblings and opinions. Rightly so. However, I must admit, with all modesty that my rumors, at least, have been better than 90% accurate. If I could translate that kind of accuracy to the stockmarket I would be a rich rumor monger. As it is we just hang in there, enjoy greatly our work getting the magazine out each month and talking to hundreds of you on the telephone each month.

I have on occasion made suggestions to manufacturers and advertisers (some being both) concerning what I actually know about our marketplace and some gut feelings I get, especially after reading about a thousand letters a month, on varied subjects, from you. Not all your letters and calls have been exactly complimentary to various vendors or products, but in every case, as many have written back to report, we did get most all complaints squared away without too much hassle. I must honestly say, despite my personal bias, that I believe that we have, on the whole, the best bunch of manufactures and dealers, of any computer group. For that small percentage that did not 'get their act reasonably straight' we just denied any additional advertising. In most cases this was satisfactory, however, in one case we were forced to go to court to uphold our right to require advertisers to honor their advertising, to the letter. Some we have no doubt missed but I can only act on what I receive that is not 'sour-grapes' and is detailed sufficiently for me to discuss it with the other party. And especially your letter of complaint must be an original (no copy machine output) **SIGNED by YOU!**

As concerns the suggestions and 'gut' feelings; well I need your input as to what you might want. Let me know, and please bear in mind that I can not answer all your letters, but all are carefully read and worthwhile suggestions are noted and relayed to the necessary parties. Believe me most manufacturers and dealers are very interested in what you, **those who are spending your hard earned cash**, want and are or are not willing to fork over for.

RUMORS

As for rumors this month, here goes. First Southwest Technical Products will soon offer a new CRT terminal that talks. This will be a split screen and keyboard unit with lots of other goodies.

Radio Shack will probably come out or at least release info soon on some new products.

I want to insert here that in all cases I try to verify what is placed here as 'rumors', which are exactly that. In this case I had Bob Nay contact a high official at Tandy to confirm some of the following, little information was given by this particular person. Sometimes the folks at Tandy are real cooperative, other times, not much. This was one of those 'not times', anyway here is what I hear on the 'grapevine'.

Radio Shack has several new disk controller designs running. One in particular using the new WD single chip. The price should be about $139.76 and it just may carry a part number like AX9060, for the color computer. I suspect that it will be announced around July 5th, before you receive this issue but after this is written. Also there are a couple of newer and fancier color computers (actually about three running) that could be prototypes for future offerings. I hear that they are sorta waiting to see what the future brings from the competition. One version just might be like a model III with about 65K RAM and built in disk drives, gotta wait and see. You are gonna be surprised when you see what popular disk system it will probably be running!

Rumor has it that a fellow down in Australia has converted the Hitachi 6809 computer to run FLEX™, if so he has done better than a lot of others I know. It has some 'lackluster' built in due to the Motorola disk controller it uses, and is not actually as flexable (no pun intended) as the Radio Shack color computer. Have a promise of some details coming, let you know (when).

Seems that Canon, Hitachi and some other UJCM (Universal Japanese Computer Manufacturers) just do not understand the US market and especially why their machine should be able to run existing software. The most important aspect of any computer, is the availability of APPLICATIONS SOFTWARE! I can remember Canon trying to demonstrate STYLO on their machine at the NCC last year, it was a disaster. STYLO is a fine piece of software but it, as most all other 6809 software, has to have the proper hardware and firmware support, it didn't. On all Standard S50 Bus machines, STYLO runs like a champ. No one in his or her right mind is going to lay out hard earned cash if they have to write all their applications software inhouse, it just ain't necessary! There is too much GOOD applications software already up, debugged, and running. Also it does not cost an arm and leg to buy (look over this issue). In addition the cost of developing brand new, inhouse software is far, far above the cost of buying software that has it's cost spread out over many users. Not to mention the hours of debugging, patching and all the other frustration that come with virgin software. Nowadays most computer prospects insist on good, dependable and inexpensive applications software. Gone are the days, for the most part, for a heavy demand for development software, some still desire and need it, but the crop of current new users do not.

When the Japanese and even some domestic manufacturers wise up to this then we will really see 6809 equipment sales boom. If you cannot transport it and run it on most 6809 computer, then it is doomed to be a slow getter. Just look at the guys who sell lots as opposed to those who, for various and sometimes unknown reasons, struggle to stay in the marketplace. It just boils down to this, most folks are going to do it in the easiest and least painful manner available! I don't care what it is.

One manufacturer once came to us with a unit for advertising approval. Yes, Virginia we do require it. He had a fine product, only one catch, his engineers had botched the MRDY line interface, might work sometimes but not often. We brought this to their attention (after they had produced 1,500 units at a cost of over $400,000.00!) They had to patch the entire production, what a shame. Guess who eventually paid for this blunder?

New Publication Soon (?)

While at the SWTPC distributors meeting last April the subject came up concerning a need, from those running the larger (UniFLEX - OS9 Level II - Business Machines). Seems they wanted a newsletter to make available the most recent patches, fixes (from original vendor as well as users developed) and other tid-bits that could make life a little easier in the operation of their equipment using the larger or more complex 6809 disk systems.

Since then I have tried to 'feel out the water' so to speak. Seems that most of the users want this service but also seems that some of the suppliers feel that they are getting their information out on a timely basis, and as such a newsletter is not necessary. So there we sit!

Now I am going to leave it up to you guys, if you want it then CPI (publishers of 68 Micro Journal) will do it, but lets all get off on the right foot. If one side or the other does not wholeheartly cooperate, it just won't fly. Despite the fact that we (68 Micro Journal) cover more of the 6809 market than all the other magazines combined, if there is not enough support, from both sides, forget it. This means that if you have something that will benefit the other users, you have to let us know, so we can pass it along.

It is anticipated that this will be a newsletter published as info is gathered. No particular mailing date. Could be that with items of extreme importance, I would have to get out **special IMMEDIATE mailing,** thats going to be expensive. Also I hope that it will not require advertisers to continue, time will tell. Also it will not be fancy (slick magazine type) but will be accurate and to the point (most likely no editorials either) but well worth what it cost.

Cost being a nebulous thing the rates will be something on the order of $38.50 a year to start. This may not hold up long but I will accept the initial ones at this rate until I can determine what a fair price is. Also I will secure some helpful utilities to return to those who submit material or give subscription extensions, whichever works best. Actually your cooperation is going to determine a lot of the above, if not all. So if interested let me know SOON!

Now see what 'ramblings' mean. Thanks to all who keep me informed, even if we do miss the mark occasionally. Keep those cards and letters coming.

We try...

Occasionally I receive, mostly from newer readers, suggestions that we improve our operation, in one manner or another. I always welcome and appreciate your input concerning a better magazine. However, as for the level of communication, text and content wise, I feel that we do it just about right to reach and be understood by most of our thousands of readers. Especially since I must consider that we have a readership that spreads from the beginning novice user to highly technical scientist at some of the world's most advanced facilities. Not to mention the large number of scholarly educators that read 68 Micro Journal each month.

In view of this broad spectrum of readers we must adduce the content as would appeal to most. That being the case, the following is our stand on text styles for 68 Micro Journal:

"In promulgating our esoteric cogitations and articulating our superficial and real sentimentalities, amicable, technical, philosophical and psychological observations, I strain to beware of platitudinous ponderosity. Eschewing all conglomerations of flatulent garrulity. Dejeune babblements and asinine affections. I pledge always that our extemporaneous decantings and unpremeditated expatiations have intelligibility and voracious vivacity without rhodomontade or phrasmical bombast. Sedulously avoiding all polysyllabic profundity, sittacious vaculty and ventriloqual vapidity!"

Having once and for all time (I hope) dispelled the myth that we cannot expound on a level with some of the newer and self proclaimed 'better' magazines. I restate to you what I first said four years ago - "We shall, as plainly as we can, bring you more 68XX information each month than all the others combined, and in a manner such that all can understand!" That holds just as well today as it did when we first started.. You remember - way back there when there was NO magazine and hardly any articles for you the 68XX user.

Today - August 1982 - 68 Micro Journal still has more meaningful articles - for the 68XX user than all the other computer (any kind) magazines combined!!

Not to mention that 68 Micro Journal reaches more 68XX users, from industry to hobbyist, than all the other computer (any kind) magazines combined - **ANY!!**

If you don't believe it - then ask some of the readers and advertisers who have 'wasted' their dollars on other so-called "68XX" magazines. I know, and think that you should also. We are not the 'slickest' but we are, by far, the BEST 68XX magazine and intend to remain number **ONE!**

In fact we just may be "the only real 68XX magazine (any kind).

FIRST 68XX USERS MEETING

The first 68XX users group organizational meeting was held in Atlanta, Georgia June 12th and 13th June 1982. The first meeting was held Saturday evening at which time a general discussion was held and temporary officers were elected. Russell Gorr and Mark Sproul were nominated and elected to head up the initial organization. Also the official name "SIG68 GROUP" was chosen.

In all there were over 100 68XX users present and it looks as if the organization has gotten off to a good start. It has been my contention for some time now that we needed such a group. This should foster a closer tie between the users and the manufacturers and dealers. I was disappointed (as were most of the users attending) that there was not a better representation of our manufacturers and dealers. Of the major manufacturers the only one attending was Richard Don of GIMIX. I had indicated to some of you that there would be others attending, but for various reasons (unknown to me yet) they did not show. Richard had just completed a hard week at NCC, yet he was interested enough in the users to come, and bring his lovely wife Arlene, to Atlanta. We all appreciate his sincere interest and contribution to this initial get together of 68XX users. Also attending were folks from Terminus Designs, Southeastern Micro, System-68, Data Systems 68, Computer Systems Consultants, Alford and Associates, Data-comp/S.E. Media and of course 68 Micro Journal. Also attending were Steve Odneal who did the Data-Comp version of FLEX for the color computer and Scott Schaeferle who is the author of Dynacalc. Users from as far away as Germany and Canada attended as well as hundreds from across the country. I expect to see SIG68 Group grow in the next few years.

The aims of SIG68 Group are the fostering of a closer relation between the users and users and suppliers. Also the collection and dissemination of software (non-copyrighted - public domain) such as is done by various other computer groups. A library using the FLEX format will be available to members for a small copy fee.

Initial operating funds were donated by 68 Micro Journal until the group gets established. Also 68 Micro Journal was happy to host the first two meetings in Atlanta. It is anticipated that each year the meeting will be held in a different part of the country, in order that all members and users will have an opportunity to attend. Next year the meeting is expected to be held in Trenton, New Jersey.

Also announced at the meetings was information concerning two new Computer Bulletin Boards. One will be at the SIG68 Group headquarter (New Jersey for now) and one also at 68 Micro Journal.

The 68 Micro Journal system will be on line within the next few weeks (if all goes well). The system will require an identification code (a number that is on your mailing label) to access. The format will be FLEX and the following commands will be allowed:

BUILD.CMD - LIST.CMD - DIR.CMD - CAT.CMD

These should allow sufficient latitude to send or receive text files and send or receive messages. The software was written by Mark Sproul who designed the Thomas Instrumentation Modem card. Mark also hooked up all the various hardware parts into a working computer. The baud rate will be 300 baud only. A special program will

monitor all files entered, to screen for profanity. Any member sending a text file to the system containing profanity will be barred from any future access! We hope that the younger users can access this system without that type of garbage. More on this including the telephone number will be published very soon.

The system is made up of donated parts from the following manufacturers and dealers - GIMIX 6800 CPU and memory cards - Thomas Instrumentation Modem Card and video board - Febe mainframe - SWTPC disk system, Video Monitor and other assorted boards (donated by Data-Comp/S.E. Media . This is strictly (for the time being) a system for the use and pleasure of 68 Micro Journal subscribers. Later on we may include files for passing and receiving communications with 68 Micro Journal office personnel.

Any dealer, manufacturer or reader is invited to contribute to this system, software or hardware, as appropriate or available. Also we hope to have available a general modem program that will allow our readers access to the system at a modest cost, if any of you have such software and desire to donate it we will make it available on disk, and if sufficient interest warrants we will try for a tape version also.

One thing that we all must be very particular about is the passing of copyrighted text files to the system. If this is abused it will require us to end this service! I will not knowingly allow unauthorized software on the system, we owe it to those who have made things better for all of us not to 'steal' another man's work.

More next month.

DMW - - -

Flex User Notes

Ronald W. Anderson
3540 Sturbridge Court
Ann Arbor, MI 48105

TWO WAYS OF SOLVING A PROBLEM

I have a couple of observations this time, and then we will get down to the business of the month. First is that 90% of writing a program is done in the first 10% of the time. Maybe it just seems that way. I recently wrote a program for an instrument. It receives data from a couple of vibration transducers and performs calculations on that data to produce a result. I wrote the main program including all the calculations and tested it with data from a dummy data file on my development system over a weekend. About 20 working days more of programming to get the program to accept data on the basis of interrupts from an external device, drive a digital display, and run in its own hardware were required to finish the job.

Perhaps not everyone finds this to be the case. I have a good friend who works in a manner totally different than mine, and though we work fairly well together, his method thoroughly irritates me, and the inverse is also true. This friend likes to sit down in front of a piece of paper (for either hardware or software design) and explore all the possibilities very carefully, working through worst case timing problems in the case of hardware, or trying a dozen ways to program a particular procedure before he dives in and starts programming. I know, that is what the "authorities" teach, and maybe the method I use doesn't work for everyone, but I get farther faster thinking about a problem when I have some attempt at solving it in front of me, working or not. Therefore, I generally devise the simplest solution to the problem that I can envision, build it, and find out why it doesn't work. Sometimes, I've found that that simplest approach does work with a patch or two for things I've overlooked, and I think I end up with something simpler than might be arrived at by the other approach.

In terms of time, I suspect either approach averages out to taking the same amount. Please note that I'm not advocating completely random approaches to a program or design. I ordinarily attack the problem I think will be the most difficult first. Having an approach to that, I fit it into all the easier parts of the solution, trying to keep the overall design organized. I think most people fail at writing a program or designing a circuit because they don't take the time to find out why their first attempt didn't work. Yes, sometimes I find a flaw in my first try that is unfixable because I've overlooked some factor completely, and that necessitates starting over again.

I suppose if I had to describe my usual approach to a programming problem, I wouldn't call it "Top Down" and couldn't call it "Bottom Up", but would have to settle for "From the Middle Out".

JPC A/D CONVERTER BOARD

Some time ago I reported having purchased a JPC analog to digital board and having had some success using it. I would like to amplify on those original comments a bit, and point out the remedies to some difficulties I had with it. First of all, I tried running the BASIC program that comes with the card to try it out. All seemed to work perfectly. Maybe I'd better back up and describe the card. It uses a National Semiconductor ADC0817 chip. The chip contains a 16 channel multiplexer and an 8 bit A/D converter. The chip is interfaced to a standard Motorola PIA (6821). One port is used to read the 8 data lines of the 0817 to access the data resulting from a conversion. The second port is used to supply the chip the channel number for the multiplexer. Since some lines were left over, JPC provided an amplifier with programmable gain. The output lines that program the gain are connected to the control leads of a 4016 CMOS switch.

My application required use of Assembler software to drive the A/D, and when I had it running, I noticed what seemed to be interaction between input channels. I discovered the cure after some trial and error programming. It seems that you turn the "convert pulse" on by writing $3C to the control register, and turn it off again by writing $34 to that address. The sample program provided by JPC does that with successive instructions. You first must set up the channel to be converted by writing to the proper register. I found out that the 0817 doesn't actually switch channels when you write in a new address, but when the convert pulse goes high. It starts converting when the convert pulse goes low. A look at the data sheet shows that the ALE (address latch enable) input is separate from the convert command pulse input. JPC has tied the two together, which looks to me to be no problem if the pulse is held on long enough for the address and the multiplexer output to settle. JPC has added an amplifier between the Multiplexer output and the analog input to the converter, which adds the settling time of the amplifer to that required for the multiplexer, indicated on the 0817 data sheet. Apparently, the signal wasn't settling between selecting the channel and starting the conversion. The cure was to put a short delay between turning the convert pulse on and turning it off.

```
      LDY #$20
DELAY LEAY -1,Y
      BNE DELAY
```

This nicely took care of the channel crosstalk. It may be much more delay than required, but I have lots of time in my application, so I played it safe. Now, I noticed that conversion was "rock solid" for values near the ends of the range, but quite unsteady for mid range values. My oscilloscope showed that the amplifier was oscillating when the signal input voltage was near midrange. I was using a programmed gain of unity. A look at the 4016 data sheet in the RCA CMOS data book shows that it is not a very good switch when run on a 5 volt supply. In

fact, the switch impedance becomes very high for an input voltage of about 2.5 under those conditions. Since I was using a gain of 1 continuously, I removed the 4016 and closed the feedback path with a piece of wire. The oscillations stopped and readings were steady for all input voltages within the range of the A/D. I later removed the amplifier as well, and connected the multiplexer output directly to the signal input of the converter.

I have a couple of suggestions if you have one of these and want to improve its performance. The simplest is to disconnect the Vdd (+power supply) pin of the 4016 from the 5 volt supply and connect it to the unregulated 8 or 9 volt line. The 4016 doesn't exhibit the high impedance problem when run on more than 7.5 volts. A simpler solution might be to remove the 4016 and plug in a 4066, which is pin compatible, and is a better switch. I've not tried either of these cures since my problem was solved by removing the switch and the amplifier.

You may run into one other problem. The A/D is designed to run with the 5 volt regulated supply on the board as a reference. An input of zero volts will produce a digital output of 0. An input that matches the reference produces an output of 255. If you happen to be converting an amplified signal, you must be careful to limit the amplitude of the input signal to within the range of 0 to the reference voltage. Exceeding these limits will cause crosstalk in the multiplexer. If you find the range of 0 to +5 volts rather unhandy as I did, you can add the resistor network shown below to change the input range to -10 to +10 volts. Of course, you can change the range on as many or as few input channels as you need.

You may have an application in mind for the A/D that only requires a couple of analog channels, as I did. It occurred to me that I could connect switches to the other channels and switch either 0 or Vref into the inputs. Those extra channels may then be used for "sense switches" to pass your application program additional information such as how to handle the input data, which routine to run, etc.

I wrote JPC a letter concerning the oscillation problem, and they tried (once) to call me at my office one day when I happened to be out. Apparently they don't write letters, as I haven't heard from them regarding my (or their) solution to the problem. Let me quickly add that this is in no way intended to knock the product. I've designed the A/D board into a commercial instrument, and I will be using the JPC boards regularly. I have reached the conclusion that the folks there don't quite know just what they have!

Confirming that judgement is a call I received last night from a reader who is trying to use the board. His words, as nearly as I can remember them were, "It works just fine with the BASIC program, but readings are not steady with the assembler drivers supplied." I asked if he had seen crosstalk between channels, and the response was positive. I suggested the delay (lengthening the convert pulse). This reader indicated that "there are probably lots of users out there who are having the same problems. You ought to write the solutions up in your column." This reader had also called JPC to ask about the problem and indicated that they were apparently not aware that there is a problem. Perhaps 99.9% of the purchasers are using the BASIC version of the drivers and of course BASIC has enough delay built into the interpreter so that the problem doesn't show up.

NEW ASSEMBLER

So what else is new, you might ask. All we really don't need about now is another assembler. Right? Wrong. This one, written by Frank Hoffman, is very special. I'm talking about the new CRASMB available from Frank Hogg Laboratory. Why is it special? Simple. It is 99.95% compatible with the "standard" TSC 6809 Macroassembler. The difference is that it is a cross assembler. This one is very nicely done. The syntax and assembler Pseudo operations are all the ones we are very familiar with. The assembler has associated with it a number of "personality modules" that will make it (as of now) an assembler for 6809, 6805, 6800, 6502, 8080, Z-80, or 1802. What's more, operation is automatic. The source program contains a single line (sort of a library call) that tells the Assembler what processor module to load. That is about all there is to it. An updated version that just arrived, allows specification of the processor module on the command line, so you don't have to edit your old assembler source files at all to use this assembler.

As a test, I added the processor selection line to my program MULTI from last month's column, and assembled it with no other changes. There are a few very minor differences in the format of the listing produced, and one subtle difference in the command line. The options are the same as the TSC assembler's, and are invoked in the same way, +BNGS or +LSY for example. The difference is that you may not use a comma or space before the +. If you do use a separator (which the TSC Assembler allows, but to my surprise does not require) the options are all ignored. On the disk with each processor module, are a number of test files, one for each processor for which there is a "personality" module. The test file is a group of assembler instructions that use each instruction and addressing mode for that particular processor. That's a rather nice way to verify that the assembler does work properly for all of the processors.

I have an old KIM-1 board around somewhere. I could never get the ambition together to fire it up again after having gotten used to using a good assembler with the 68XX. It would be easy to write a "download" utility to send it programs in the format it expects from a tape. It would also not be difficult to bypass the cassette interface and go directly into the serial input. That means that I can go back to all those old issues of Kim-1 User Notes and type in all the programs. Sort of makes me want to get my old Tiny BASIC out and run it again!

If you work with processors, and sometimes use a "foreigner" for a project, you can now use your SS-50 machine as a development system for all the processors mentioned above. If you don't like the download option, you could always program EPROM's to plug into another system. I assume that the list of processors will be extended to include the 16 bit ones eventually. Perhaps the 68000 will be included. Now, Frank, all we need is a series of emmulators for those other processors so we can write programs and test them on our systems. That would be great for learning Assembler on some other processor! For pricing, see Frank Hogg Laboratory ad. Oh, by the way, I had to stretch hard to find something to complain about in this package. The version I received for evaluation doesn't print the system date on the header line of each page. I mentioned that one to Frank Hoffman who called me about something else recently, and he indicates that printing of the date is now being incorporated in an updated version. What would be even nicer in addition to that feature, would be to have an option in the command line that would cause the assembler to prompt for the time, and input that to be printed on the header line of each page too! (I'm just kidding, Frank.)

MORE ON ASSEMBLER MULTIPLY PROGRAM

Last month I promised to continue the discussion of the multiplication routine, and to add the necessary code to make the routine do signed arithmetic. I am going to defer the complications of doing conversions from ASCII input to BINARY and the inverse for a little longer. We can use the command line approach again. We are going to add the requirement that you input the sign of each argument on the command line, ie. the number will be

preceeded by a '+' or a '-'. That will simplify the problem a bit so we can concentrate on handling signed arithmetic without getting bogged down in unnecessary details. You will therefore test the routine by using: SMULTI.BIN.1,+12,-17 for example, as the format. If you omit either sign, the result will be 0.

Flex has a subroutine called NXTCH. This routine will get the character pointed at by the command line buffer pointer. It happens that this pointer, (Flex BUFPTR) is pointing at the character after the separator (space or comma) after the filename, when the program starts. If that character is an ASCII + or a -, we only have to test to see which. The added code is simply:

```
        JSR NXTCH       TO GET THE SIGN
        CLR SIGN,U      INITIALIZE TO POSITIVE RESULT
        CMPA #'+        SEE IF POSITIVE
        BEQ GETONE      SKIP NEXT CODE IF POSITIVE
        COM SIGN,U      RECORD FACT THAT SIGN IS NEGATIVE
GETONE  JSR INDEC       THIS IS PART OF OLD CODE
```

This code is repeated as the second argument is obtained from the command line. If both signs are negative, the two COM SIGN,U operations result in SIGN being cleared (positive). If either sign was negative, SIGN,U is complemented only once, and it is $FF. SIGN,U therefore represents the sign of the result, zero indicating positive, and non-zero negative. Now the multiply is done as in last month's code. When we get to the output portion at the label OUTNUM, we simply add a test:

```
        TST SIGN,U
        BEQ OUT2
        LDA #'-         IF NEGATIVE OUTPUT THE SIGN
        JSR PUTCHR
OUT2    LEAX NSLB,U     PART OF OLD CODE
```

I've not only put off the ASCII BINARY conversion problem, but the handling of the NEGATION operation too. Since numbers are generally input as sign and magnitude, we can test this routine just as it is. Perhaps last month I didn't say it directly, but obviously we could extend this double byte multiply to as many bytes as we like. Of course as we increase the number of bytes, the routine gets more complicated rather quickly. We would probably want to code the names of the bytes for the result as B1, B2, etc.

I can hear some of you saying "That's cheating. You didn't do any signed arithmetic at all, just handled a bit more information." Let me assure you that since the 6809 MUL instruction is an unsigned multiply, it has to be done that way. For those who just can't wait a month, if you have your 16 bit integers stored as two's complement numbers, you simply test the sign, use the instruction COM SIGN,U to save the fact that it was negative, and then negate the value. Suppose the value is in the D register. You would code the following:

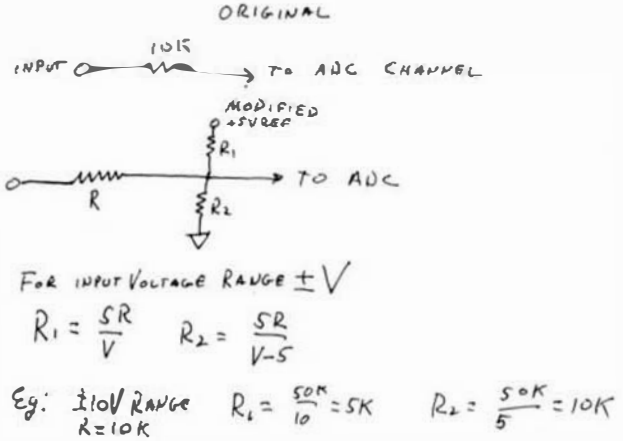
```
        LDD VARIAB
        BMI NOCOMP
        COMA
        COMB
        ADDD #1
        COM  SIGN,U
NOCOMP
```

If you didn't want to code that twice, you could make the test and negate a subroutine.

Next time, we will look at addition and subtraction, again inputting our arguments with INDEC and NXTCH. That means that we will have to convert negative input values to two's complement form, and convert negative answers to sign and magnitude as we just did in the above code. We will go into more detail next time.

```
MULTIPLY PROGRAM                5-11-82 2:15 AM 6809 PAGE   1

                CD03  WARMS   EQU   $CD03
                CD48  INDEC   EQU   $CD48
                CD39  OUTDEC  EQU   $CD39
                CD1E  PSTRNG  EQU   $CD1E
                CD24  PCRLF   EQU   $CD24
                CD27  NXTCH   EQU   $CD27
                CD1B  PUTCHR  EQU   $CD1B
                      *
                      INIARG  MACRO
                      ARGS    SET   0
                              ENDM
                      *
                      DEFARG  MACRO
                      \1      SET   ARGS
                      ARGS    SET   ARGS+\2
                              ENDM
                      *
                      * LOAD ADDRESS 0
                      *
0000                  START   INIARG
0000                          DEFARG H1,1     HIGH ORDER BYTE OF ARG 1
0000                          DEFARG L1,1     LOW ORDER BYTE OF ARG 1
0000                          DEFARG H2,1     HIGH ORDER BYTE OF ARG 2
0000                          DEFARG L2,1     LOW ORDER BYTE OF ARG 2
0000                          DEFARG MSB,1    MOST SIGNIFICANT BYTE OF RESULT
0000                          DEFARG NMSB,1   NEXT MOST SIGNIFICANT BYTE OF RESULT
0000                          DEFARG NLSB,1   NEXT TO THE LEAST SIGNIFICANT BYTE OF RESULT
0000                          DEFARG LSB,1    LEAST SIGNIFICANT BYTE OF RESULT
0000                          DEFARG SIGN,1   SIGN STORAGE
                      *
                      * MAKE ROOM ON THE STACK FOR THE VARIABLES
                      * THEN POINT U AT THE VARIABLES SO THAT THEY MAY BE
                      * REFERENCED WITHIN ANY SUBROUTINE IN THIS SECTION OF CODE.
                      *
0000 32  77                   LEAS  -ARGS,S
0002 1F  43                   TFR   S,U
0004 BD  CD27                 JSR   NXTCH     GET A CHARACTER FROM THE FLEX INPUT BUFFER
0007 6F  48                   CLR   SIGN,U
0009 81  2B                   CMPA  #'+
000B 27  02                   BEQ   GETONE
000D 63  48                   COM   SIGN,U    MAKE IT $FF IF ARG 1 IS NEGATIVE
000F BD  CD48  GETONE  JSR    INDEC   GETS ARG 1 IN 1 REGISTER FROM COMMAND LINE
0012 AF  C4                   STX   H1,U
0014 BD  CD27                 JSR   NXTCH
0017 81  2D                   CMPA  #'-
0019 27  02                   BEQ   GETSEC
001B 63  48                   COM   SIGN,U    RESULT 0 FOR ++ OR --, FF FOR +- OR -+ ARG SIGNS
001D BD  CD48  GETSEC  JSR    INDEC   GETS ARG 2 IN X
0020 AF  42                   STX   H2,U
0022 CC  0000                 LDD   #0
0025 ED  44                   STD   MSB,U     CLEAR RESULT AREA
0027 ED  46                   STD   NLSB,U
                      *
                      * NOW MULTIPLY
                      *
0029 A6  41                   LDA   L1,U
002B E6  43                   LDB   L2,U
002D 3D                       MUL             LOW ORDER TIMES LOW ORDER
002E ED  46                   STD   NLSB,U    MEET TO THE LEAST SIGNIFIANT BYTE
0030 EC  41                   LDD   L1,U      A GETS L1 AND B GETS H2
0032 3D                       MUL             GET A HIGH AND LOW ORDER 'CROSS PRODUCT'
0033 E3  45                   ADDD  NMSB,U    MEET TO MOST SIGNIFICANT BYTE
0035 ED  45                   STD   NMSB,U
0037 24  02                   BCC   MULT1
0039 6C  44                   INC   MSB,U     CARRY IF NECESSARY
003B A6  C4    MULT1  LDA     H1,U
003D E6  43                   LDB   L2,U
003F 3D                       MUL             OTHER HIGH AND LOW ORDER 'CROSS PRODUCT'
0040 E3  45                   ADDD  NMSB,U    SAME POSITION AS LAST 'CROSS PRODUCT'
0042 ED  45                   STD   NMSB,U
0044 24  02                   BCC   MULT2
0046 6C  44                   INC   MSB,U     HANDLE CARRY IF NECESSART
0048 A6  C4    MULT2  LDA     H1,U
004A E6  42                   LDB   H2,U
004C 3D                       MUL             HIGH ORDER TIMES HIGH ORDER PRODUCT
004D E3  44                   ADDD  MSB,U
004F ED  44                   STD   MSB,U
                      *
                      * NOTE THAT IF BOTH ARGUMENTS WERE WITHIN RANGE SUCH THAT
                      * OVERRANGE COULD NOT OCCUR, THE RESULT OF THIS LAST MULTIPLICATION
                      * WOULD HAVE TO BE ZERO.
                      * NOW WE CAN TEST FOR OVERFLOW.
                      *
0051 1083 0000                CMPD  #0
0055 27  09                   BEQ   OUTNUM
0057 34  00 001B              LEAX  ERROR,PCR POINT AT ERROR MESSAGE
005B BD  CD1E                 JSR   PSTRNG    PRINT IT
005E 20  11                   BRA   EXIT      RETURN TO FLEX
0060 BD  CD24  OUTNUM  JSR    PCRLF
0063 6D  48                   TST   SIGN,U
0065 27  05                   BEQ   OUT2
0067 86  2D                   LDA   #'-
0069 BD  CD1B                 JSR   PUTCHR
006C 30  46    OUT2   LEAX   NLSB,U  POINT AT RESULT FOR OUTDEC
006E BD  CD39                 JSR   OUTDEC
0071 32  69    EXIT   LEAS   ARGS,S
0073 7E  CD03                 JMP   WARMS     NORMALLY WOULD HAVE RTS HERE
```

```
                E
0076 20 4F 56 45  ERROR  FCC   / OVERFLOW - ARGUMENTS TOO LARGE./
007A 52 52 4F 4F
007E 57 20 2D 20
0082 01 9E 07 55
0086 4D 45 4E 54
008A 53 20 54 4F
008E 0F 20 0C 41
```



COLOR User Notes

Robert L. Nay
5900 Cassandra Smith Rd.
Hixson, Tn. 37343

This month we'll do a little "house cleaning" by looking at several different items, from running 64K to some new items announced for the Color Computer.

First, there is some confusion about the 64K Modifications; what is entailed in the Modification, what Computer configurations can be Modified, etc. Basically, ANY Version of the Color Computer can be Modified; some are easier that others. If you remove the Top Cover of the Computer and look on the edge of the PC Board right beside the front edge of the Cartridge Slot, you will find a Number-Letter showing the "PC Board Version". The "Letter" designation determines how easy it will be to Modify. The latest Boards are "E" Versions, which have been available since around Christmas. "E" version Boards only require setting up the Jumpers in the Computer and connecting one of the spare NOR Gates in the Computer to allow activating the additional 32K RAM. Earlier Version Boards will require cutting PC Board Lands on the bottom of the Computer Board and adding a few jumpers in addition to the above NOR Gate Mod. These take the place of the Radio Shack installed jumpers on the "E" Version Boards. The earlier the Board, the more cuts and jumpers required. "D" Version Boards are not bad, "C" Version Boards require additional changes, and earlier Boards should be taken back to Radio Shack as they will probably be replaced.

Also, you **MUST HAVE a Version 1.1 BASIC ROM** (not the EXTENDED BASIC ROM). You can check the Version Number of your ROM by entering **EXEC 41175**; this will cause the display of the regular BASIC ROM's logo, giving the Version Number with it. The V1.1 ROM is standard on the "E" Board Computers. It is required because the V1.0 BASIC ROM does not look for the 32K jumper (remember, when it was programmed, 16K was the maximum amount of Memory you could get for the Color Computer). If you install 32 or 64K RAM and turn the Computer on with a V1.0 ROM, every command you give will result in an **OM ERROR** (Out of Memory). You do NOT have to have EXTENDED BASIC to run 32 or 64K Memory.

There have been several articles relative to the 64K Installation in various magazines, but none are complete within itself. **'68' Micro Journal** will have one by Clay Abrams either in this issue or the next one. Modification instructions are included with **DATA-COMP**'s 64K Mod. Kit for the "C", "D", and "E" Boards.

Once you install GOOD 64K RAM Chips, you can run the BASIC Program in Frank Hogg's Article in the April '82 issue of **'68' Micro Journal** to get an idea of whether the Memory is good or not. When you operate the Computer, it is just like the Radio Shack 32K System. The 64K is **REQUIRED** to run either the **DATA-COMP** or **Frank Hogg Labs** FLEX System because FLEX requires RAM from $C000 thru $DFFF (and the Systems also use some RAM above FLEX). You can also experiment with the Memory Map Type 0 paging capabilities of the MC6883 SAM Chip, which effectively divides the 64K into two 32K pages which can be selected thru software (ref. the Motorola MC6883 Data Sheet). If you use a BASIC POKE Command to switch the SAM to the Memory Map Type 1 (64K RAM Mode of operation), the Computer will go into limbo because you have turned off all of the ROM's and have no Keyboard or Display Routines left in the Computer. Both FLEX Operating System Conversions load these Routines into Memory above FLEX, so you have communication with the Computer when the routines are loaded into Memory from Disk.

You MUST have EXTENDED BASIC to run the Radio Shack Disk System. Many of the Disk BASIC Commands are in the EXTENDED BASIC ROM, and the DISK BASIC ROM is accessed through the EX. BASIC ROM.

We received a new MC6883 SAM Chip from Motorola the other day. This Chip runs somewhat cooler in the Color Computer. It seems a large part of the heat problem with the Chip is due to impedance mis-matches with the address lines, and this is an updated version which seems to have eliminated some of the problem. At least the problem is not being ignored; hats off to the Motorola troops.

We also received the **Frank Hogg Labs** FLEX Conversion for the Color Computer the other day. The Documentation with the package is excellent, and consists of additions to the regular TSC FLEX Manual. As stated in an earlier issue, we have asked Ron Anderson, who writes the FLEX Users Notes Column, to review the two packages (so we could provide an unbiased view for the readers), so I won't go into details about the system. The basic differences between the systems are as follows:

1. **Frank Hogg Labs** provides a **PUTBOOT.LDR** utility which is used in making up a "System" Disk. This utility must be used with a newly initialized Disk and is used to put the programs on the Disk in such a way that the Disk can be used as a "Boot" Disk. It actually configures a few Tracks on the FLEX Disk in the Radio Shack format and installs a special Boot Routine there, so that when **RUN"FLEX** is entered after the Computer is turned on, it finds the Radio Shack portion of the Initialization routines and switches to FLEX at the proper time to get the system "up and running". This allows the use of a System Disk just like the normal FLEX Users are used to using. The DATA-COMP System requires two Disks; one to use for "Booting" FLEX, and after you are in the FLEX Operating System, you remove that Disk and install a normal System Disk. The one problem is that **PUTBOOT.LDR** can NOT be copied from the furnished Disk, so the furnished Master Disk must be used any time a new System Disk is to be made. Normally, this will not present a real hardship, because you NEVER use the Master Disk (of ANY program) for normal use anyway; it should be filed away for safekeeping as soon as you get a few System Disks made up.
2. A **SETUP.CMD** is used to set up various options within the FLEX System, such as Disk Drive configurations (i.e., Single/Double Sided, stepping rates, Drive locations, etc.), Printer configuration (baud rate, add line feeds or not, stop bit count, etc.), a Memory option which provides a small memory examine/change monitor capability, and a Terminal option (which allows control of the TV Display such

as blinking or non-blinking cursor, a green or orange display background, control of the "bell" feature, etc.). The "Bell" is a nice touch; this is a short tone output through the TV Speaker which is used in programs as an error warning, to let you know that a long procedure is finished, etc. The SETUP.CMD can be included in the "STARTUP.TXT" file (as I did on my system), or can be used to "patch" FLEX permanently with the APPEND.CMD. Normally, you would use the APPEND.CMD for functions which will not change, like Printer configuration, while the STARTUP.TXT would be used for those that may change once in a while (for instance, the Terminal configuration). The DATA-COMP System does not presently have this feature; some, such as the Printer configuration, Disk stepping rates, etc., can be APPENDed to FLEX, the Terminal features are not available (Franks' initial release has much better Terminal Controls than DATA-COMP's, although DATA-COMP will have them in their updated version by the time you read this, just like we understand that Frank Hogg Labs is working on the increased Video Display routines similar to those of DATA-COMP's 42x24 and 51x24 Displays).

3. The Frank Hogg Labs conversion does not provide the increased Display Screens, it uses the normal 32x16 Radio Shack Display with inverted colors for lower case etc. (as stated, I have heard he is working on them).

These are the major differences between the systems; both have good points, so pick your poison. The **Frank Hogg Labs** FLEX Conversion will run FLEX Programs just like the **DATA-COMP** Conversion, and vice versa. The **Frank Hogg Labs** FLEX System sells for $99.95 and does not include the Editor and Assembler; the **DATA-COMP** FLEX System sells for $199.95 (General FLEX plus F-MATE(RS)) which includes the Editor and Assembler.

Finally, are they compatible; i.e., will each read the others Disks? Yes and No; they ARE compatible for Single Sided Disks, both Single and Double Density, but the FHL System can not read the back side of the DATA-COMP Disk on **TRACK 00**, which is a continuation of the Directory. It does fine until it needs to read past Sector $0A on Track 00, then nothing. Confusing; let me give you the rest of the story. First, GIMIX is happy as a lark with any configuration disk from either system; Single or Double Density, Single or Double Sided. On the other hand, SWTPC can only read Single Sided, Single Density Disks from either System reliably, and MOST of the time it can read Double Sided Single Density; but try Double Sided, Double Density, and it can't find the Front Side of the Disk. Finally, you can NEWDISK a Double Sided, Double Density Disk on SWTPC, and usually read it with no problem with the DATA-COMP FLEX; FHL's FLEX can't find one or two Sectors every other Track or so; it varies. But, it cannot find the SAME Trk/Secs on any one Disk; i.e., it misses different Trk/Secs on different Disks, but it can't find the same identical Trk/Secs on any one Disk **every time you test that Disk!!!!** So, **ALL SYSTEMS are compatible SINGLE SIDED, SINGLE DENSITY**, which is the FLEX Standard for 5 1/4" Media transfer, and that is the IMPORTANT thing. If you are exchanging Disks, make it SS, SD and there is no problem anywhere.

In summary, there is little to choose between the two Systems. The **Frank Hogg Labs** FLEX Conversion for the Color Computer is well done with many well thought out features, and comes with excellent documentation.

The MICRO WORKS (P.O. Box 1110, Del Mar, Ca. 92014 -- 714-942-2400; see Adv. this issue) has announced a couple more products for the Color Computer. Andy Phelps, the Software "magician" who gave us such excellent products as the **SDS80C** Assembler Rompak, the **CBUG Monitor**, and still the only **Color Computer Disassembler**, has a couple more "goodies" coming. The first is the first Radio Shack Disk based Assembler I am aware of, and this one will be a full-blown **MACRO ASSEMBLER** with an excellent Screen oriented Editor and a Debugging Monitor. The other item that also looks extremely interesting is a **FORTH Language** Rompak. This unit should really make Programming on the Color Computer interesting. FORTH is a "different" Programming Language that is relatively easy to learn and extremely powerful. It offers some of the best features of what might be called the "Ideal" programming language in that it is completely interactive, like BASIC (i.e., you can Execute a statement directly from the Keyboard to check out a routine, etc.), and yet it compiles easily to an extremely fast running program. This should be a real winner on this Machine, with special Graphics commands, etc. We'll present a complete report on these two products as soon as they arrive; based on past experience with the other products **The MICRO WORKS**, I would suggest you get your orders in early, because they will be outstanding products.

We just now obtained a copy of the long-awaited Radio Shack Editor/Assembler Cartridge, **EDTASM+**. I haven't had a chance to really work with it much yet, but I am very impressed with the **ZBUG** Monitor that comes with it. This Monitor has several unique features that are really nice when trying to figure out why a Program is not working as it "obviously should", such as the capability of examining memory either within address ranges OR within program LABEL ranges. You can also disassemble the code in either HEX or MNEMONIC mode, and change memory using either form of command. We'll look it over and give a full report next month.

Finally, a lot of the new FLEX Users have asked how to get the Control-C (or <BREAK> Key on the Color Computer) to stop an XBASIC Program Run. If you refer to the TSC Extended BASIC Manual, in the Section on ADAPTING TO YOUR 6809 SYSTEM, page 81, they discuss the User Supplied Break Routine. They start off by saying "If your system does not use an ACIA for terminal input, you will need to supply a routine which checks to see if a character has been received from the keyboard.". Yep, that's us; no ACIA. Then things get VERY UNCLEAR. If we don't want the break capability, the ACIA address should point to a zero byte to disable it; since we DO want the capability, evidently it must point to a NON-ZERO byte. Therefore, the first thing to do is insure that the ACIA address (location $4D-4E, per page 80) points to something that cannot ever be zero. How about the RESET Vector; that will never be zero while we are running FLEX and XBASIC. So, we set $4D-4E to point to a "non-zero" byte. Next, we take the three addresses they give us on page 81, and fill in the x's with our own routine to complete the patches.

```
ORG $4D       make the ACIA address point
FDB $FFFE       to a "non-zero" byte
SPC
ORG $22       choose the "reaction" to a
FDB $8DE8       Control-C or <BREAK> (see below)
SPC
ORG $015C     fill in our own "check keyboard"
JSR [$E21C]     Routine ($E21C contains a pointer
FDB $1212       to the routine, so use INDIRECT)
SPC
ORG $0220     same thing, different location
JSR [$E21C]
FDB $1212
END
```

A few notes here; the $8DE8 at location $22 gives us a halt when we hit <BREAK>, then the program will resume if we hit another <BREAK>, or stop and return to READY if we hit any other Key. If you want the original <BREAK> to stop the program and return to READY right then, insert $8DEB instead of $8DE8. (Magic?? No, this information was in an earlier Manual.) Next, notice that we only need two NOOP's ($12) as fillers because the JSR Indirect produces a total of four bytes of code instead of the normal three bytes shown in the example for a regular JSR ($BD). Finally, there is no reference after the Program's END statement. We don't want any "transfer address" here, since this is just a "patch" to the XBASIC Program. Now all we have to do is Assemble this program and Append it to XBASIC. We will call this "masterpiece" **BASICPAT.TXT**, so the following should make XBASIC act the way we want:

ASMB BASICPAT,LS (to stop the listing on the Display)
APPEND 0,XBASIC.CMD BASICPAT.BIN XBASIC.CMD

Assuming you have the Drives set up as System=0 and Work=1, you will now have the new XBASIC.CMD on Drive 1. Try it out, (go into XBASIC with **1,XBASIC** <cr>), and if all goes well, copy it over to the System Disk.

--- RLN ---

F-MATE

TRS-80C DATA-COMP F-MATES(RS) REVIEW

By: Clayton W. Abrams K6AEP
1785 Comstock Lane
San Jose, Calif, 95124

When Don Williams contacted me and proposed a Flex review for the TRS-80C, I was rather excited at the prospect. I have always been aware of the possibilities of the TRS-80C and felt FLEX was a natural evolution to bring this low cost system into the real world of big time computing.

To give a little background I have been running Flex for the past three years on my SS-50 system. I have used SSB DOS and CP-M and it is my personal opinion that FLEX is one of the best disk operating systems available for any microprocessor.

The main selling point for FLEX is its simple command structure and easy to remember syntax. Other disk systems rely on the complex use of semicolons and colons to partition the various fields which is not very friendly. I rarely ever have to consult the FLEX manual to use the utilities, the syntax is standardized and easy to figure out unlike CP-M.

At the point in time when Don Williams contacted me I was not equipped with the necessary hardware in the TRS-80C and I had the computer updated to 64K and disk within a few weeks. After obtaining the Tandy disk controller and adding 64K RAM I awaited that moment when the system could be booted up.

But before this can be done you must obtain two software packages, a TRS80C with 64K, regular(ver 1.1) and extended basics, and some drives with controller. One way of obtaining a disk controller board is buy one from Tandy. Some discount stores sell the controller separately or one can be obtained from Tandy National Parts (P/N AX9060). * Also from - DATA-COMP Div, CPI (see advertising this issue), at a discount! In this way you can use any existing 5 1/4 drive which you may have. The software can be obtained from DATA-COMP in two parts. Since I already had the General FLEX9 package it was not necessary to obtain this software. If you already have the General FLEX9 it is only necessary to obtain the F-MATES(RS) software from DATA-COMP.

The F-MATES(RS) package consists of two disks and 28 pages of instructions. To load this FLEX system requires a few simple steps. Step 1 is to first insert the F-MATES(RS) installation disk in drive 0. Next type the Basic statement: RUN "FLEXLOAD", a machine language program will be executed. After a few seconds a message will be displayed on the screen asking to insert the FLEX disk. Next remove the installation disk and install TSC General Flex. You next hit a key and the FLEX.COR will be loaded and you will see the three plus signs on the screen. This is all that is required to bring up the TRS-80C FLEX for the first time.

This moment was very exciting and opened a whole new world for this little computer. It's hard to imagine how much work it took on my big SS-50 system to reach this stage. The first time I brought Flex up on an old SWTPC 6800 system involved weeks of work and the purchase of a disk controller card with a price tag greater than my base TRS-80C.

The F-MATES package includes a excellent documentation package which makes the whole installation job very simple. In the documentation more information is provided on making bootable disks and even some methods of producing a half TANDY DOS/FLEX DOS formatted disk. This speeds up booting of the system. Once the system is booted you can view and experiment with all the utilities available with general Flex and the F-MATES(RS) packages.

THE UTILITIES

The F-MATES package comes with a selection of utilities on the second disk. In the following I'll give a brief run down of the various utilities. Five of the utilities are used to format the TV screen. Their description relates to the characters per line and the lines per screen. They are; V32X16, V32X24, V42X24, V51X24 and V64X24. These utilities are actually overlays to a common display routine and all but the 64 X 24 appear to be satisfactory in appearance. I did put a mod into the TRS-80C to extract composite video. On a Sanyo 9 inch black and white monitor the 64 character line appeared to be marginally acceptable. To obtain composite video all that was necessary was to connect the TV through a capacitor on pin 1 of the RF modulator in the TRS-80C.

Two additional utilities pertain to memory and disk modifications and are: 1. DISKEX.CMD - This utility is very similar to the EXAMINE Utility in standard Flex. The utility allows displaying and modification of disk sectors on a Flex disk. The utility functions with all video formats. 2. MEMEX.CMD - This is a utility which allows for the modification of memory, and is useful as machine language monitor while still in FLEX.

The remaining utilities are an assortment of various functions. One of the more useful in the remaining F-MATES is USERKEYS.CMD. This utility allows any one of 12 user defined keys. The user keys can contain any hex code. These user keys can be accessed through FLEX by use of the CLEAR key. For example if you had entered a 4F into the userkey 0 position to output a 4F all that is necessary is to type CLEAR then then 0.

The package would not be complete without the NEWDISK.CMD utility which is used to format new virgin disks. The utility is similar to the TSC version with a couple of major exceptions. Either single or double density disks can be created. A very nice feature is that any number of tracks can be initialized. The F-MATES instructions provides a method of initializing the lower 15 sectors as a Flex disk, and the remainder as a Tandy disk. This flexible feature of this version of NEWDISK has some very interesting possibilities.

Another interesting utility is the RSDIR.CMD which is used to display a directory of a Tandy formatted disk while in FLEX. I can't wait till someone developes a file transfer programs to and from Tandy DOS and Flex. Probably by the time this is published it will be available.

A very clever utility which opens a large number of possibilities is SAVEROM.CMD. This utility copies the Tandy BASIC ROMS to disk. When this is done it is possible to run Tandy Basic programs while Flex is resident and operational. The only restriction is to first format the video screen to 32 characters on 16 lines. Some interesting possibilities are to: 1. Create a version of Tandy BASIC for your own applications, 2.

Correct some of those BASIC problems which bug many of us i.e. Audio on/off or the OK prompt. 3. Transfer machine language programs from Tandy cassette tape to a binary files on FLEX. When in BASIC to return to FLEX all that has to be done is type FLEX.

SYSTEM CUSTOMIZING

As with all programs I never seem to be happy until at least one modification is made. With the F-MATES overlays I found one small objection which is the stepping rate of the drives. The drives which come with a standard FLEX system have a stepping rate of 30 MSEC per track. Since my drives are MPI-B51 they have a band stepper the track to track rate is 5 MS. After disassembling the FLEX drivers, I located a table which contained a byte which is transferred to the controller to preset the stepping rate. The table contains a byte for each drive and contains four bytes of 03 starting at location $DE5B. If these bytes are changed to 00 the drives will step at a 6 MSEC rate. I made this patch into a series of FCB's and assembled it into a binary file. I then modified the STARTUP.TXT file in FLEX to GET the binary file. FLEX is then modified and the drives will step at a faster rate.

FLEX compatibility

Since I have had this program package I have found few standard FLEX programs which will not function on the TRS-80C. I can't say that I have investigated all possibilities since my software library is not that large. Generally any program which uses FLEX calls to subroutines will function in this environment.

Some of the programs which I have verified are: TSC Basics, Text Processor, Disk Utilities, Standard utilities, and PASCAL. The only problem which I noted was with TSC Basic. This problem was that a control C was not recognized as a stop processing character. I understand from Bob Nay that this condition will be corrected in later versions of the program. Other programs which I have verified is Brian Bailey's (WB4MMP's) Filesort and a local Silicon Valley Ca version of a small C compiler.

One thing which should be remembered when running FLEX on a TRS-80C is not to press reset on the rear of the computer. This will cause the color computer to reset to standard BASIC and it will be necessary to reload the Flex system.

CONCLUSIONS

It's my opinion the DATA-COMP adaptation of the FLEX System for the TRS-80C presents one of the best values in home computing available for any low cost computer to date. It's hard to believe that a $299.00 computer can be expanded upwards by adding RAM, ROMs and disks to such a powerful system at such a low price.

Editor's Note: In the interest of fairness I would add that the Frank Hogg (see advertising this issue) version of FLEX™ also runs well on the Color Computer. Our primary interest is that FLEX becomes the 'other' disk standard for the Color Computer from Tandy. This opens up a vast pool of software that can be run on the Color Computer and also directs the Color Computer user in the proper direction when he/she decides to "upgrade" to a better or larger system.

C O L O R C L I N I C

TONY DISTEFANO
4680-18 Street
Laval West DUE
Canada H7R 2P9

A month has gone by already and it's time to write again. Before I go on, I would like to mention again, that my goal in this column, is to help you with the hardware side of your computer. This means, to solve any problems, from bad memory to no picture at all. In your letters, try to be specific with your problem, listing all the syaptoms, when and how often it happens, and the make and model of the peripherals if any that are involved. I am also interested in your hardware projects. If you have a project and you are interested in sharing it with others, jot down a short description of what it does and how much it would cost to build. Send it to me. I'll write it up in this column and if interest warrants, I'll include all your details on how to put it together.

Personally, I have a proto-board that I put together. It connects to the Color Computer via a modified Radio Shack program Rom-Pac. On this proto-board I put together an Eprom Programmer. The programmer itself cost me about four dollars, that's the price of a PIA and a LATCH chip. It costs a little bit more if you don't have a 20 to 30 volt power supply. The proto-board itself is under $30.00, not counting the price of a ROM-PAC, less if you have a used parts box. OH! by the way this programmer can program the new 64K (8K x 8) chips. If you are interested in seeing this project in detail drop me a line and I'll start it.

Anyway that's enough for the future, now for the present. Since there is about two months delay between my first article and your letters I'm on my own again this time. No letters to answer. That's no problem because it gives me room to talk about the Motorola MC14529 the dual 4 channel analog multiplexer.

This chip has the duty, in the Color Computer, of routing all the analog signals to their proper places. It has two outputs, one of which is the output from one of the four joysticks (2 sticks by 2 dimensions). Depending on the two select lines one of the four joysticks will appear as output. This output then goes into the plus input of a voltage comparator (1/4 of U14 LM339). The other side of the comparator comes from the D to A output. The putput of the comparator goes to PA7 of U8 (MC6821). PA7 is the last bit in port A of this chip. This bit is programmed as an input by the Basic initialization routine.

The other output on the MC14529 is connected to the AF (Audio Frequency) input of the RF (Radio Frequency) modulator. This is the sound that you hear from your TV's speaker. Only three of the four inputs are used, the other is shorted to ground. The three inputs are as follows : 1- The six bit D to A; Used to simulate music or other sound effects. 2- Connected to the output side of the cassette player; This mode gets activated with the "AUDIO ON" command. Whatever sound is coming from your cassette player will come out of the TV's speaker. 3- Connected to one of the pins of the cartridge connector. This allows an external sound generator or synthesizer to be routed to the TV's speaker.

The output to the TV can be one of these three sources, or the output can be deselected. In this case the output of the MC14529 becomes high impedence. Now comes the interesting part. When Radio Shack started making the new 1.1 models, they decided that this high impedence state was prone to background noise, so they added a terminating resistor. A 300k ohm resistor to ground was added to the output of the MC14529. Well it ended up causing more noise than if removed. When they found this out, they immediately stopped putting this resistor in. In the meantime who knows how many computers they sold with this resistor in it? Well I got one of those computers when I upgraded to the 1.1 version. If you are one of these people that get a loud buzz when you play the Radio Shack CHESS, CHECKERS or POLARIS then you have one of the computers that have the resistor. To eliminate this problem you have to cut out this resistor.

The resistor is R81 on the new Technical Reference Manual. It is located just to the right of the RF Modulator and above the RF shield. Cut it out completely and that should eliminate that buzz.

Since this chip connects to the outside world of the computer via the joysticks and cassette port, it's very vulnerable to exterior elements like static electricity or higher than rated voltages. Although every effort has been made to protect this chip it still can be permanently damaged. If for some reason your joysticks don't work or you don't get any sound from your TV, you may have a blown chip. Replace it with a good chip and that should do it.

I found another problem with this chip. If you find that every now and then you lose sound to the TV when you do any I/O to cassette, it's the MC14529 chip that is defective. Change it. In general if you have any problems with sound or joysticks, try that chip first.

Next month a real treat. How to get green text with a black background for under five cents. That's right FIVE CENTS. It's a hardware modification that fouls the VDG (Video generator) into thinking that uppercase characters are lowercase and vica-versa. Only ASCII characters are affected not graphic characters. This does not change any color modes at all only text.

DUB

DUB review

by
David R. Lewis Pres.
Mid South Pharmaceutical Inc.
P. O. Box 584
Hixson, Tn. 37343

A UniFLEX assembler language program for decompiling TSC UniFLEX BASIC.

DUB (Decompiler for UniFLEX Basic) runs under TSC UniFLEX and will create source statements from programs compiled with TSC Basic. DUB will allow you to recompile all old programs in newest versions of Basic. Another advantage of using DUB is the old "this program doesn't quite do what I wanted syndrome" is not as big a problem anymore. DUB allows you, the user, to take your compiled programs, examine the source, make your changes, and recompile. Another problem DUB solves directly involved me, if I might give an example.

I had written a program to print a special packaging label for our company, on a plain white label. Afterwards, we ordered preprinted labels with our letterhead. By the time we received the preprinted labels, and had used up the old, I had somehow lost the source to my label program. I needed to remove the letterhead part of the program. Luckily DUB came around just in time and the fix was a cinch. After having experienced this, I now believe anyone with a lot of compiled programs, especialy those who keep their source copies in a separate location, should have this utility available just in case of emergency.

You're probably asking about now, just what exactly does this thing do? Well DUB will decompile programs compiled under all versions of TSC UniFLEX Basic and Precompiler. DUB will decompile a program compiled under version 1.04 allowing you to recompile under version 2.06. DUB is very operator oriented and easily leads the user through each step of operation, of which there are only 2 or 3. Using the command "dub" calls the decompiler which then prompts for the pathname of the compiled program. Next it prompts for output desired; ie,(1-Terminal 2-Disk); in case of disk output the last prompt is for source file pathname. The newest version of DUB works like any other command. For example: for a listing (dub compiled.bac) or for editing (dub compiled.bac source.bas) assuming for this example both the compiled and piped to file are in the current directory.

DUB operates in a flash and will list the source of a compiled program, as far as I'm concerned, as fast as the list command will list a regular source listing. In case of disk output it is ready for editing and/or recompiling.

Some of the features of DUB are that if the compiled version is from Basic 1.04 or compiled by the Precompiler, then DUB assigns variable names starting with example (aa,aa$,aa%) next going to (ab) and so on. While assigning names DUB checks to make sure it doesn't assign any invalid variable names such as (as, as%, as$, if, fn, ect.).

If the programs were compiled under Basic 2.04 or later, then DUB uses the same variable name as was originaly coded. Making changes and debugging even easier.

One thing that I noticed in using DUB is the difference between different versions of Basic including the Precompiler. Programs compiled using the precompiler or 1.04 are a little harder to work with because the varible names are different from the original source. Maybe soon the Precompiler, which I like using so well, will store the varible names in the compiled version.

In closing I would like to give some words of praise for Mike Williams who wrote the DUB routine. Mike impressed me very much with his programming abilities and knowledge of the UniFlex system. His help was very beneficial to my deeper understanding of DUB, and was greatly appreciated.

STOCK RPT

John P. Tucker
12015 O'Connor Road 184
San Antonio, Texas 78233

STOCK PORTFOLIO REPORTER, a software package for the 6800 system.

This is a review of a very excellent piece of software that runs exactly as advertised, performs each of its functions without flaw, and can prove to be of interest and assistance to users who are attentive to the financial world of today, even if in a minor way. It is a necessity to any holder of a reasonable portfolio of marketed stock.

The program is "Stock Portfolio Reporter," written and distributed by Micro Investment Systems; 411 Longleaf Road; Shreveport, Louisiana 71106. Menu driven (a blessing for this type of program) and heavily disk oriented (but requiring only a single disk drive, although two are handy), the program system is written in a dialect of Basic. More about that later, if there is room. Basic protocol is used throughout and is so thoroughly prompted that the user needs almost no computer expertise to use all of the facilities offered. The program is well-polished and apparently completely debugged.

The only computer-oriented tasks left to the user are to get Basic into memory, load the Main Menu (LOAD MAINMENU and RUN), and then follow the prompts.

The Main Menu offers four choices: Entry and Update; Report Display; Just Starting; and End. Ending at anytime during use of the program is done through the Main Menu in order to clean up the open files, etc.

The "Just Starting" item is a clever condensation of the beautifully written manual. It explains in enough detail the "why and when and how" aspects of the system to allow almost anyone to proceed with the program. The items of information are presented one at a time on the terminal screen. The

item lengths are relatively short and are displayed clearly for study and comprehension. Since they are replaced only at the operator's command, unlimited time may be used for perusal.

Ordering Entry and Update from the Main Menu brings up a second menu of seven segments. "Enter the Portfolio" allows just that: one inserts his present portfolio of stocks along with such pertinent information as number of shares, price paid per share, a market index at the time of purchase (I use the Dow Jones average but any applicable index may be applied), and date of purchase, as well as some other minor bits of information as would be anticipated. This one-time entry is the basis upon which the program operates.

Since portfolios do not usually remain static, provisions are made in the Entry and Update Menu to "Add a New Stock," "Remove a Stock," and "Correct Stock Information (e.g., Dividends per Share)." These titles adequately explain the purpose of the sub-programs. Each is completely prompted and works without those despicable error reports from Basic.

From the Entry and Update Menu, one may also print a list of the stocks in the portfolio, allowing detailed examination of all of the portfolio information being retained on disk. As is usual with this particular dialect of Basic, any properly configured port and printer combination may be used for the print-out.

Because market prices are anything but fixed, this group of menu items contains an "Update Market Prices" selection. This information must be newly input each time the programs are run in order for the system to properly report upon the present condition of the portfolio.

Naturally, there is a selection to return to the Main Menu, also.

The beautifully prepared manual (I cannot be too complimentary about this manual) "holds your hand" through each keystroke you must make to run the system of programs. Not a single entry is skipped over by the manual. You need know nothing more than how to press the keys on the terminal. The manual even tells you when to use the Return key! Still, it is not "talking down" at any time.

Having input the portfolio and caused the stock list to be printed to your satisfaction, a return is made to the Main Menu to select the Report Display section. Here you are presented with a sub-menu of five items: Price and Yield Report, Price Action Report, Price and Yield Summary, Price Action Summary, and the usual Return to Main Menu selection.

It is assumed, by this writer, that the holder of a portfolio of stocks will understand each of the terms used in the previous paragraph. To attempt to define and explain them would be much too lengthy for a review here in the pages of our Journal.

Nothing is perfect, so a word of warning is inserted at this point. The report print-outs are NOT configured for a typical eighty column printer. This is the major objection I have to this set of programs. Apparently it was written with a Teletype Model 43 printer in mind using the 11 1/2 wide paper. Fortunately, I had some 15 inch paper on hand for my Epson MX-100 and thus had no trouble getting my reports. Discussion is underway with the very friendly author of this program about several minor changes I would like to see made, this being the main change.

The neatly printed reports (with nicely justified columns of decimal figures) are informative and, as best we can determine, very accurate.

Again, the manual explains each and every move to be made.

Being written in Basic, the programs take a while to operate. This is especially true since the version I have on hand is written in one of the older "slow" Basics distributed by SWTPCo.

Apparently at one time, SWTPCo "tampered" with TSC's various versions of Flex, marketing in many guises something they called Flex 1.0. My version of Flex 1.0 resided entirely in low memory — $0000 to $7FFF with the DOS residing from $7000 to $7FFF. TSC once informed me that the proper name of this version was miniFlex and that the 1.0 designation was "something SWTPCo tagged on there". I also have a version of Flex 1.0 purchased directly from TSC that is the equivalent of Flex 2.0 but for 8 inch disks, with the DOS residing in high memory!

Now I know of still another version of "Flex 1.0"! This one appears to be "miniFlex" that has been moved to reside from $A000 up, leaving more programming room in memory below $7FFF. The DOS is for 5 inch disks. The Basic available to this version was labelled Version 3.0 and appears to be the same dialect as was used with the original miniFlex but with the End of Memory set up higher, leaving adequate room for the Stock Portfolio Reporter to run unhindered (it won't run in my miniFlex version because of End of Memory errors encountered).

You should be aware that you will need this particular version of Flex and Basic to run the programs.

Being very disk oriented, and being limited by the dialect of Basic to sequential files, there are some "healthy" pauses during compilation of some of the reports. Don't bother to make your coffee before starting the programs; you'll have time to do it during the runs. (I couldn't find the place in this version of Flex to change the disk track seek speeds. Had I been able to take advantage of the high seek speed capability of my new drives from Data-Comp, the programs would have executed appreciably faster.)

I can truthfully recommend these programs to holders of modest portfolios. They are accurate and informative and very helpful. The manual is the best of any software we have reviewed to this time. The author seems to be anxious to help with any difficulties encountered (our difficulties being limited to getting the right operating system and Basic), and the programs seem to run without fault.

This review of the 6800 programs makes us anxious to see a version written for TSC's Extended Basic (with random files) to run on both the 6800 and 6809. What a delight it would be to have this up and running on the 6809 Color Computer! Here is a chance for some really marvelous graphics.

"C" User Notes

Norm Cosmo
3 Pryor Road
Natick, MA 01760

Compilers for all C'sons

With that awful pun, we start a review of the major C compilers on the market today. To put things in perspective bear in mind that "today" is early June. If I don't get them out soon, the vendors will be looking for a noosing in Natick!

Some ground work. There are two flavors of C compilers available, "real" compilers and "Small C" compilers. This choice of titles is not meant to belittle Small C derivatives. It is meant to show their heritage. What distinguishes them? I can only give you my definition.

I apply the term "real" to compilers such as Bell Lab's, Whitesmith's, and Introl's that present the whole language or most of it AND are faithful to the language standard as defined in Appendix A of "The C Programming Language" by Kernighan and Ritchie.

The Small C compilers are modeled after Ron Cain's article in Dr. Dobb's Journal. They are usually characterized by the fact that they lack many of the languages features and may deviate from the standard by varying degrees.

The Small C compilers can still be quite good as we shall see. But they do fall short of being a full blown compiler for the whole language. This of course it reflected in their price which ranges from one tenth to one fifth what some of the "real" compilers cost. You get what you pay for.

Besides a review of the actual package, I will include some code produced by the compiler. The code will be for the simple function streq() which is passed two pointers to strings and returns TRUE if the strings are the same, FALSE otherwise. It is shown below.

```
#define TRUE 1
#define FALSE 0
#define NULL 0

streq(s1, s2)
   char s1[], s2[];
{
   int i;

   i = 0;
   while (s1[i] == s2[i])
      if (s1[i++] == NULL)
         return(TRUE);
   return(FALSE);
      }
```

The C source will be listed in the compiler's output code so that you can get an idea how the compilers handle each statement.

INTROL -- version 1.1

The Introl C compiler is the only real C compiler available and being shipped today. It is the best of the bunch if you are after the real thing. The package consists of the following

compiler PP.CMD the preprocessor
P1.CMD the parser
P2.CMD the code generator

A09.CMD A two pass absolute assembler. it is TSC compatable except for macros which it does not support.

MLINK.CMD A one pass linker that takes object modules and libraries and builds an executable binary.

MLIB.CMD A program that allows the user to maintain a library file of object modules.

STDLIB.LIB The standard library containing most of the commonly used C functions and the runtime code.

STDIO.H . The standard header file, which has a variety of standard defines.

FLEX.H A header file of FLEX dependant definitions.

Also included is a test program, plus you get the C source code for all the functions and the assembler code for the runtime packages and a few of the standard functions that can't be done easily or efficiently in C. Quite an impressive package, which is what you would expect for software in the $300 range. The only real hitch is that it is just available on 8" disks.

The compiler is a dream to use. You create a source file using any of the standard editors available today. To invoke the compiler you would type the command line

```
+++PP FILENAME
```

This would invoke the preprocessor. It produces two files from your source code. The first is a called a string file and contains all the string that were encountered in the file. The second is called a lexeme file. This is a compressed, binary representation of the source commands with all the appropriate macro substitutions and lexical analysis performed. PP will automatically call P1 if no errors have occurred.

P1 is the parser. It takes the lexical tokens (lexemes) and generates what is know as a parse tree, doing some very minor optimization in the process. Its output is another binary file called the parse tree file. Yet another form of the source program. It automatically chains to P2 if no errors have occured. When your source codes gets past P2, the rest of the trip is more or less guaranteed. If any errors occur during PP or P1, they will ask you if you want to continue or stop.

P2 is the code generator and optimizer. It takes the parse tree file and produces 6809 assembler code. The code is position independent except for jump tables to external function references. P2 then chains to the assembler A09 which assembles the code in two passes to produce a relocatable object module. The second pass of the assembler ends the compilation process.

If the file just compiled was the main program or a major portion of one, then you might next want to link it to any other modules or libraries and test it. Or, if the file was itself a library function, you would most likely use MLIB.CMD to put it into one of your libraries.

Introl did something rather clever. Some where in the compilation process relocation tables are built into the output file. These are tables that use the common assembler directives such as FCB, FDB and FCC to describe the object module. These tables are then used by the linker to pull in all the needed functions and resolve all the various references, finally yielding an executable ".CMD" file. When using the linker, you specify the object files and libraries that you want it to scan. It will determine whether the named files are objects or libraries and treat them accordingly. Additionally, there are commands that let you ORG the program where you want it and determine which of the three runtime packages to use.

These runtime packages each include all the 16 bit arithmetic code, and do the following.

$RUN -- Initializes the machine state as well as the globals and statics.

$RUNLINE -- Does all of $RUN and also parses the FLEX command line arguments into valid C strings. it then stacks a count of the arguments as well as a pointer to an array of pointers to the strings before entering main().

$RUNFULL -- Does all of $RUNLINE and also allows you to redirect the terminal I/O to

and/or from files. $RUNFULL is the default runtime option.

MLINK is a one pass linker. This implies that a call to a function must occur before the linker encounters the actual code for the function, so it will know to pull it in. Where this affects you is when you are building your own library, or modifying the standard library. You must order, by position within the file, all the functions in the library so that all their references to other library functions are also forward references.

The librarian makes this task very easy. MLIB.CMD allows you to creat new library files or modify existing ones. The only limitation is that it doesn't work from disk to disk, so the library must be able to fit in a memory buffer. There are commands that allow you to manipulate the modules within the library. For example you may add a module to the buffer, or replace a module in the buffer with a file of the same name. You may insert a new module into the middle of the buffer, or delete a module from the buffer. There is a command that list all the modules in the buffer in order. Finally, you can save the buffer back to the disk.

The documentation is well done. Each phase of the compilation is thoroughly explained. The manual goes into great detail about the object and library file structures. There is a section on how the compiler uses the 6809 registers and how you can write your own assembler code for use with the compiler. Finally, they are very clear in distinguishing the differences between their version of C and the language as described in "The C Programming Language" by Kernighan and Ritchie.

The only features of the language not supported are initializers, longs, floats, doubles, and bit fields. I have listed in the order that I would like to see them added, but I know that floats, doubles and longs will be the first additions, since that will open up a lot of business applications for the compiler.

The code produced by the compiler is compact and fast. Since it is a compiler, there will be some inefficiency, but it's not excessive. In fact I found it fun to read the the compiler's output file and figure out ways to force the compiler to generate more efficient code.

There were a couple of bugs that I found in version 1.0. One was a problem with stack overflow. Some types of parsers have a problem with error recovery. Certain types of errors would really confuse the compiler. It would "get out of sync" with the source code and generate a long list of errors. It would also use up some more stack with each error until the stack finally ran into the code and the system crashed. This bug was fixed by limiting the maximum number of errors. When that many errors are found the compiler just quits. This is a nice feature that saves you a lot of time since you don't have to wait for the whole program to compile. These bugs were fixed with version 1.1.

68 Micro Journal reader Marty Poppe uncovered a few bugs with the runtime math routines, and one or two bugs with the way the compiler set up the code for mathematical expressions. He told me that in most cases he would call the bug in to Introl and would usually get a return call very shortly thereafter with the fix. These bugs should be fixed by release 1.2 which should be out in late June.

I must give the Introl C package an AAA rating. The package works smoothly. The manual is very good. Finally, known bugs have been fixed quickly. Introl seems quite bent on producing one of the best packages on the market. The Introl compiler is my "working" compiler that I use for all my program development.

The Introl compiler does not allow you to imbed the C source code in the assembler file since the code is generated from a parse tree, not the original C program. In the following listing the source was imbedded by hand. The actual code exists between the labels .streq and Jtab, all the rest is part of the relocation info. Note that the Introl compiler allows better label definitions than the TSC assembler. The compiler has the following register assignments

D -- the primary register (B for chars)
Y -- the pointer to the global and static variable storage area
S -- system and variable stack
X, U -- general usage

```
        fcb     2
        fcc     /TESTFUNC/
        fcb     0
        fdb     mend-mstart
mstart
        fdb     cend-cstart
        fdb     cstart
        fdb     0,0,1
        fcc     /streq/
        fcb     0
        fdb     .streq-cstart
cstart
.streq:
* int i;
        leas    -2,s
* i = 0;
        clrb
        clra
        std     0,s
* while (s1[i] == s2[i])
T1.4
        ldd     6,s
        addd    0,s
        pshs    d
        ldd     2,s
        ldx     6,s
        ldb     d,x
        cmpb    [,s++]
        bne     T1.23
* if (s1[i++] == NULL)
        ldd     0,s
        addd    #1
        std     0,s
        subd    #1
        ldx     4,s
        ldb     d,x
        cmpb    #0
        bne     T1.4
* return(TRUE);
        ldd     #1
        leas    2,s
        rts
T1.22
        bra     T1.4
* return(FALSE);
T1.23
        clrb
        clra
        leas    2,s
        rts
Jtab
cend
        fdb     0,jtab-cstart
        fdb     0
mend
        fcb     3
```

WORD'S WORTH -- version 1.0

The Word's Worth compiler is probably the best of all the "first version" Small C derivatives. The compiler represents a fairly big subset of C and includes the data types char and int, the complete control structures with the exception of the GOTO along with single index arrays and pointers. The package includes

CC.CMD — The compiler itself.

COPT.CMD — A single pass optimizer that works on the output file of the compiler. The source, in C, is also included.

RLOAD.CMD — A pseudo-linker (my title) that has been described in previous issues of 68 Micro.

MACRO.TXT — A special macro header required by the code produced by the compiler to make relocatable object modules when assembled.

TEST.C — A program to test the compiler.

The function library and runtime code exists as a series of ".LIB" object files and their sources. There is a test program, PRIMES.C, with its accompanying ".LNK" file, which will be explained later.

In use, you start a C compiler by entering the FLEX command

```
+++C
```

This invokes the compiler, which then issues the following series of prompts

```
Do you want the C text to appear [N]
Output filename?
input filename?
```

The [N] shows the default answer assumed if just <cr> is struck. If no file is entered for the output file, then the output goes to the terminal. The compiler will process input files until <cr> only is entered, at which point it cleans up and exits to FLEX. The recommended default extensions are ".C" for the source text and ".ASM" for the compiler's output.

The next step is either the optional optimizing stage or the assembly. The optimizer also prompts for the names of the input and output files. The recommended output default extension is ".OPT". it is claimed that the compiler will pare off anywhere from 15 to 35 percent of the code. I have no argument with this figure based on my usage. It seems about right.

The optimizer is itself written in C and the source coded is given to you as part of the package. Reading it was rather enjoyable. The optimizer consists of a five or six line buffer. The buffer is filled from the assembler file and scanned for certain sequences of instructions. When such a sequence is found, it is replace with a single instruction. The optimizer goes through the entire file this way, replacing the inefficient sequences that it knows the compiler produces. The idea is very adaptable to just about any compiler. I suspect that many of our readers could code one for the Dugger or Intersoft compilers rather easily.

Word's Worth suggests that the assembler be called with the command line

```
+++ASMB INFILE.OPT,OUTFILE.REL +YP10
```

The P10 option will start any listing at page 10, saving you a little time and possibly a lot of paper. The first 10 pages are the expansions of MACRO.TXT. What you get out of the assembler is a relocatable object module. It is made relocatable by the inclusion of relocation tables in a scheme similar to Introl's. Therefore the code out of the compiler is uniquely tied to MACRO.TXT, which it includes via a "LIB" at the beginning of the module. If you change MACRO.TXT, you may be grabbing a tiger by the tail.

The final executable binary is produced by running RLOAD. RLOAD will require the existence of a file called "xxxx.LNK" which is referred to as a link. This file is a list of the all the modules needed to be pulled into the program to make it complete. The give an example of a link file with their example program PRIMES.C. The contents of that file is

```
:NAME PRIMES.CMD
CHEAD.LIB
PRIMES.REL
CCHAR.LIB
CMATH.LIB
CCOND.LIB
```

The first line names the output of the link, which is the executable binary. Next MUST come the FLEX09 linkage module. Finally the program module(s) and any required runtime support and function modules.

Other than making sure that you have made up the link file properly (not left anything out), there really isn't much to it.

There is major difference between the way a linker, such as Introl's and a "pseudo-linker" such as Word's Worth's works. This difference is what leads me to apply the term "pseudo". A real linker will strip out of a library only the code needed to complete a program. A pseudo-linker will take only those modules that you specify, but is will take all the code contained a module, even if it not used by the program. Still it's very much better than no linking at all!

Recently the thought came to me of what may be optimal way to use RLOAD. What you do is to make a separate binary module of each function. Keep these and RLOAD on a "library" disk. The only thing you need on your program developement disk will be the applications modules and link file for the particular program you are working on. Since you only have to make the link file once, the extra effort to key in a few extra lines is minimal. This technique could help reduce the overall size of program down by calling in an object file for each function. Keeping them on a separate disk will also ease maintenance of the library and greatly reduce the "clutter" when perusing the program disk's directory.

in use, you would edit, compile and assemble a program using one disk. You would then "swap in" the library disk disk and build the executable binary. In fact, one of the first C programs that you might want to build is an automatic link file generator that prompts you for the ".CMD" file name and main program modules; and then emit a list of functions, letting you answer y/n for each one that you want in the program. I have not tried this, but it seems like a very useful method that is not overly cumbersome.

For a Small C derivative, the compiler has a fairly complete set of the language. The differences between Word's Worth C and the standard are well documented in the manual.

I was impressed with the quality of the manual. It is complete and well laid out. There is a nice tutorial that helps get you up and running with the compiler.

I have uncovered no bugs with the few programs that I have put through the package. The code produced by compiler/optimizer combination seems fairly efficient for this type of compiler. The simple benchmark given in C NOTES 5 shows this compiler coming in second behind the Introl compiler for both execution and code density.

The Word's Worth compiler deserves an AAA rating. if I did not have the Introl, this would probably be my working compiler for FLEX.

Two assembler listings are given for the Word's Worth compiler. The first is the output of the compiler, the second is the output of the optimizer.
You might enjoy comparing them. The Word's Worth compiler uses the 6809 registers as follows

```
D -- primary register
S -- system and C program variable stack
X,Y,U -- general registers
```

```
** * * Small-C Compiler * * *
* ad   ted for TSC-FLEX 6809
*   By Allan R. Batteiger
*    & William M. Knight
*

      LIB MACRO.TXT
      MODULE 'Small-C Compiler Module'
*#define TRUE 1
*#define FALSE 0
*#define NULL 0
*streq(s1, s2)
      ENT streq
streq EQU *
*     char s1[], s2[];
*     {
*     int i;
      LEAS -2,S
*     i = 0;
      TFR S,D
      PSHS A,B
      LDD #0
      STD [,S++]
*     while (s1[i] == s2[i])
cc2 EQU *
cc4 EQU *
      LEAX 6,S
      TFR X,D
      TFR D,X
      LDD ,X
      PSHS A,B
      LEAX 2,S
      TFR X,D
      TFR D,X
      LDD ,X
      ADDD 0,S++
      TFR D,X
      LDB 0,X
      SEX
      PSHS A,B
      LEAX 6,S
```

```
        TFR X,D
        TFR D,X
        LDD ,X
        PSHS A,B
        LEAX 4,S
        TFR X,D
        TFR D,X
        LDD ,X
        ADDD 0,S++
        TFR D,X
        LDB 0,X
        SEX
        LBSR ccEQ
        LBEQ cc3
*         if (s1[i++] == NULL)
        LEAX 6,S
        TFR X,D
        TFR U,X
        LDD ,X
        PSHS A,B
        LEAX 2,S
        TFR X,D
        PSHS A,B
        TFR D,X
        LDD ,X
        ADDD #1
        STD ,S++
** * * Small-C Compiler * * *
* adapted for TSC-FLEX 6809
*   By Allan R. Bateiger
*    & William M. Knight
*
        LIB MACRO.TXT
        MODULE 'Small-C Compiler Module'
*#define TRUE 1
*#define FALSE 0
*#define NULL 0
*streq(s1, s2)
        ENT streq
streq EQU *
*   char s1[], s2[];
*   {
*   int i;
        LEAS -2,S
*   i = 0;
        TFR S,D
        PSHS A,B
        LDD #0
        STD [,S++]
*   while (s1[i] == s2[i])
cc2 EQU *
cc4 EQU *
        LDD 6,S
        PSHS A,B
        LDD 2,S
        ADDD 0,S++
        TFR D,X
        LDB 0,X
        SEX
        PSHS A,B
        LDD 6,S
        PSHS A,B
        LDD 4,S
        ADDD 0,S++
        TFR D,X
        SUBD #1
        ADDD 0,S++
        TFR D,X
        LDB 0,X
        SEX
        PSHS A,B
        LDD #0
        LBSR ccEQ
        LBEQ cc5
*           return(TRUE);
        LDD #1
        LEAS 2,S
        RTS
*   return(FALSE);
cc5 EQU *
        LBRA cc2
cc3 EQU *
        LDD #0
        LEAS 2,S
        RTS
*   }
        EXT ccEQ
        ENDMOD
        END

* --- End of Compilation -
        LDB 0,X
        SEX
        LBSR ccEQ
        LBEQ cc3
*         if (s1[i++] == NULL)
        LDD 6,S
        PSHS A,B
        LEAX 2,S
        PSHS X
        LDD ,X
        ADDD #1
        STD 1,S++]
        SUBD #1
        ADDD 0,S++
        TFR D,X
        LDB 0,X
        SEX
        PSHS A,B
        LDD #0
        LBSR ccEQ
        LBEQ cc5
*           return(TRUE);
        LDD #1
        LEAS 2,S
        RTS
*   return(FALSE);
cc5 EQU *
        LBRA cc2
cc3 EQU *
        LDD #0
        LEAS 2,S
        RTS
*   }
        EXT ccEQ
        ENDMOD
        END

* --- End of Compilation ---
```

DUGGER'S GROWING SYSTEMS -- version 1

The Dugger compiler was the first 6 09 C compiler to be advertised in 68 Micro Journal. The subset of C handled by the compiler is smaller than that of either Intersoft or Word's Worth. Included in the package are

DGSCC.CMD	The compiler.
CLIBR.TXT	The standard library.

And, like all the other compliers, it has a demo program, that good old standby, Eratosthanes Sieve Prime Number program.

The compiler is invokes without commands from FLEX with the command

```
++DGSCC
```

It then goes through the following prompt sequence

```
Do you want C text in the output
Do you want globals defined
What is the starting label
What is the output file name
What is the input file name
```

It then compiles the input file and asks for another, if you don't enter any name, the compiler cleans up and exits to FLEX. One thing to be aware of, if the output file already exists then you must use a new name, you can't optionally delete it from within the compiler. This is somewhat of a pain in the neck as is the fact that errors aren't sent to the screen as they are encountered, nor is there an error summary given at the end of the compile. You must list the entire output file to see if there are any errors. In general, the human interface is not very well executed.

The compiler runs fairly quickly. The code it produces is not unduly inefficient. There are problems though. For example, when character variables are used in the C programs they are promoted to integers. The problem is that under some conditions they end up in the B accumulator and at other times in the A accumulator. This makes for some messy code in the function library when handling char's. Also, the first copy that I received would not execute #include properly, this was rectified by a patch. Global char variables allocate 2 bytes of storage each instead of just one. A minor point but gauche never the less.

The runtime library has it's problem too. For example, the function tolower() would add 32 to the character summarily. It did not test it to see if it was really a letter. As originally supplied, the program would start in right at main() via a transfer from FLEX. But when the program was over it would return to FLEX with an RTS, with possible varying effects. The file handling functions were especially dangerous. As supplied for FLEX, the function codes for the FMS calls were Smoke Signal's. One of the functions, I believe it was putc(), returned directly to WARMS on an error. I finally redid most of the runtime package to make it usable. That done, and knowing about its little idiosyncrasies, it became quite livable.

The manual was in pretty tough shape, as I have stated in one of the C USERS NOTES columns. It was very unspecific as to defaults for the compiler prompts, didn't give enough detail as to the differences between Dugger C and the standard. For a reasonably seasoned C user like myself, this was a nuisance. Most questions could be answered by compiling a few quick test programs and reading the output code. For the neophyte it could very easily have been a very frustrating experience. There is an example section of C programs and functions. It is quite contrary to the recommendations in Kernighan and Ritchie.

How would I rate Dugger's version 1? It is very hard for me to be fair. Dugger C was the first that I had a chance to run and I have a lot of time invested in it. Looking at it objectively, it was necessary for the compiler to get patched for it to work, I had to rewrite the runtime library, and the manual was close to a disaster. On the other hand, I had both the Dugger and Intersoft compilers and found myself using the Dugger because of code efficiencies and speed of compiling.

Here is a sample of the code from a very early release of the version 1 compiler. The register usage is

```
   X    -- Primary register
   D, Y -- General registers
   U    -- C program variable stack
   S    -- System stack
```

```
*6809-C COMPILER REV 1.0
 NAM DGSC
*#define TRUE 1
*#define FALSE 0
*#define NULL 0
*streq(s1, s2)
STREQ  EQU *
*    char s1[], s2[];
*    {
*    int i;
 LEAU -2,U
*    i = 0;
 LEAX 0,U
 PSHU X
 LDX #0
 STX [,U++]
*    while (s1[i] == s2[i])
CC2  EQU *
 LEAX 4,U
 LDX ,X
 PSHU X
 LEAX 2,U
 LDX ,X
 PULU D
 LEAX D,X
 LDB ,X
 CLRA
 TFR D,X
 PSHU X
 LEAX 4,U
 LDX ,X
 PSHU X
 LEAX 4,U
 LDX ,X
 PULU D
 LEAX D,X
 LDB ,X
 CLRA
 TFR D,X
 TFR X,D
 LDX #0
 CMPD ,U++
 BNE *+5
 LDX #1
 CMPX #0
 LBEQ CC3
*        if (s1[i++] == NULL)
 LEAX 4,U
 LDX ,X
 PSHU X
 LEAX 2,U
 PSHU X
 LDX ,X
 LEAX 1,X
 STX [,U++]
 LEAX -1,X
 PULU D
 LEAX D,X
 LDB ,X
 CLRA
 TFR D,X
 PSHU X
 LDX #0
 TFR X,D
 LDX #0
 CMPD ,U++
 BNE *+5
 LDX #1
 CMPX #0
 LBEQ CC4
*            return(TRUE);
 LDX #1
 LEAU 2,U
 RTS
*    return(FALSE);
CC4  EQU *
 LBRA CC2
CC3  EQU *
 LDX #0
 LEAU 2,U
 RTS
*    }
 LEAU 2,U
 RTS

*0 ERRORS IN COMPILATION
 END CINT
```

SUMMARY

I suspect that by this time I've made a few people unhappy with my grading. Well, I tried my best to be objective. Of course, like everyone else, I have my prejudices when it comes to programs and how I like to see them operate. I would like to point out some ground rules that I tried to apply to each of the compilers. It may ease the pain or intensify the bliss.

One of the most important features is ease of use. I am firmly convinced that the only difference between my home system and the VAX 11/780 that I use at work should be the speed and storage capacity. Programs used at home should be just as friendly as any that I am apt to use at work. I have also spent a couple of years using UNIX, and still use it today. Programs that trick me into feeling like I'm in that environment make me very happy. I give a lot of weight to how easy a program(s) is to use and how well it is documented.

Bugs per se don't get me too excited if they are of the obscure type. Some bugs are unforgivable. They should have been tested out of the package before it got to market. The obscure types can often hide for quite a while until that certain application comes along, then BOOM. I am quite aware that any extensive piece of software will probably get into the field with unfound bugs. I really don't think that will ever change. What does concern me is how a vendor responds to any reported bugs. Prompt action is generally a sign of a trustworthy vendor.

Coherence to any language standards is also a highly weighted criteria. If you buy Pascal, then you should get the Pascal language described by Wirth. If you buy C, then you should get the language as it is described by Kernighan and Ritchie. Dialects are a pain. I would sooner forgive a compiler being a very limited, but strict, subset of the language than an extensive, but dialectic, version. Extensions are ok because they can be ignored if you're really worried about portability. But changing some fundamental of the language is taboo.

Finally, I would like to thank some of the 68 Micro readers who have helped me with their inputs. Most are probably unknown to their fellow subscribers. They include Ron Anderson, Marty Poppe, Tan Bronson and Paul Walkin. They have all been patient with my many phone calls for their opinions of this or that compiler.

Joe Wicklund
801 Duchess Rd.
Bothell, WA 98011

68 MICRO JOURNAL
P.O. BOX 849
HIXSON, TN 37343

Dear Don Williams:

I'm afraid that I have bad news for 6809 fans; I have found two instructions that do not always execute as described by Motorola. Contrary to the description in the "PRELIMINARY PROGRAMMING MANUAL", both the BGE, and BLT instructions require that the Carry bit be clear in order to work correctly. Unfortunately, the TST, INC, DEC, LD, ST, and COM instructions do not affect the carry bit; so use of BGE or BLT after any of these instructions will not be reliable. The other signed branch instructions appear to work properly.

The other part of the story involves the Ed Smith 6809 version of RRMAC. There are a number of uses of BGE following a TST instruction, causing unreliable program operation (which led to the discovery). Substitution of a BPL for the BGE should result in proper operation of these instructions as TST clears the V flag anyway. The following is a list of the locations that I've discovered, along with the address (if loaded at $4C00):

1. LABEL PROCESSING V1.0, LINE 438 ($52E6)
2. OPERAND PROCESSING, LINE 354 ($5A27), LINE 653 ($5C28), LINE 722 ($5C9E), line 752 ($5CCB)
3. dir part 1 v1.0, line 575 ($632C)
4. MACRO DIR'S V1.0, line 325 ($6F64), line 639 ($71CB)
5. OUTPUT ROUTINES V1.0, line 640 ($7446), line 672 ($747D)

There you have it, if anyone has any questions I'll be happy to try and answer them. By the way, I've been talking to the people at Great Plains Computer Company, and they seem to be very competent and helpful. They've rewritten Ed's linking loader and it seems to be easier to use. If anyone is interested, I found a fairly simple way to modify Ed Smith's 6800 RRMAC to include FCB, and FDB directives for additional flexibility in setting constants.

Home Acct Prog

Part II

ERNEST STEVE WATSON
11701 ST. CHARLES BLVD.
LITTLE ROCK, ARKANSAS 72211

OR

F. DALE BRADY
7729 BRADLEY DRIVE
LITTLE ROCK, ARKANSAS 72209

```
0 REM TSEARCH.BAS
20 CL$=CHR$(27)+"E"
30 W=60
40 PRINTCL$
50 PRINTTAB(W/2-13);"Search of Transaction file"
60 OPEN OLD"1.YEAR"AS1
70 GET#1,RECORD1
80 FIELD#1,2ASTN$,2ASTY$,3ASTM$
90 Y=CVT$%(TY$)
100 M$=TM$
110 CLOSE 1
120 PRINT:PRINT
130 PRINT"Do you want ";M$;" transaction file (Y/N)?"
    ;:AN$=INCH$(0)
140 PRINT
150 IFAN$="Y"THEN180
160 PRINT"Do you want YEAR trans file, not including"
    ;M$;" (Y/N)? ";:AN$=INCH$(0):PRINT
170 IFAN$="Y"THENM$="YEAR" ELSE CHAIN"MENU.BAS"
180 OPEN OLD "1."+M$ AS I
190 ONERROR GOTO 610
200 PRINTCL$
205 IFM$="YEAR"THENPRINTTAB(W/2-15);"Search of Year
    To Date":GOTO220
210 PRINTTAB(W/2-13);"Search for month of ";M$
220 A%=0:D1%=0:C1%=0:A1=0:M1$="":P$=""
230 PRINT
240 PRINTTAB(W/8);"1.  Account Number"
250 PRINTTAB(W/8);"2.  Check Number"
260 PRINTTAB(W/8);"3.  Date (Month/Day)"
270 PRINTTAB(W/8);"4.  Payee (Discription)"
280 PRINTTAB(W/8);"5.  Amount"
290 PRINTTAB(W/8);"6.  Return to MENU"
300 PRINT:PRINTTAB(W/8);"Your Choice (1-6)? ";
    :F=VAL(INCH$(0))
310 PRINT:IFF<1 OR F>6 THEN300
320 PRINT:PRINT
330 ONFGOTO350,370,390,430,450,340
340 CHAIN"MENU.BAS"
350 INPUT"ENTER ACCOUNT #",D1%
360 GOTO460
370 INPUT"ENTER CHECK #",C1%
380 GOTO460
390 INPUT"ENTER MONTH ",M1$
400 INPUT"ENTER DAY (1-31)",D$:IFLEN(D$)>2THEN400
410 M1$=LEFT$(M1$,3)+"/"+D$
420 GOTO460
430 INPUT"ENTER PAYEE/DISCRIPTION ",P$
440 GOTO460
450 INPUT"ENTER AMOUNT ",A1
460 PRINTCL$
470 PRINT"REF  DR  CR   CK#   DATE";TAB(27);
    "DESCRIPTION";TAB(52);"AMOUNT":PRINT
480 BE?#1
490 FORS%=0 TO 4
500 FIELD#1,S%*50ASZ$,2ASTD$,2ASTC$,2ASTN$,10ASTY$,
    26ASTP$,8ASTA$
510 IFCVT$%(TD$)<100 OR TP$=""THEN570
520 IFCVT$%(TD$)=D1% OR CVT$%(TC$)=D1%
    THENGOSUB580:GOTO570
530 IFCVT$%(TN$)=C1%THENGOSUB580:GOTO570
540 IFM1$()""ANDLEFT$(TY$,LEN(M1$))=M1$THENGOSUB580
    :GOTO570
550 IFP$()""ANDLEFT$(TP$,LEN(P$))=P$THENGOSUB580
    :GOTO570
560 IFCVT$F(TA$)=A1THENGOSUB580:GOTO570
570 A%=A%+1:NEXTS%:GOTO480
580 D%=CVT$%(TD$):C%=CVT$%(TC$):CN%=CVT$%(TN$)
    :A=CVT$F(TA$)
590 PRINTUSING"### ### ### ##### \2345\
    \2345678901234567890123456\$#,###.##",A%,D%,C%,
    CN%,TY$,TP$,A
600 RETURN
610 IFERR=30THENPRINT"REENTER":RESUME
620 IFERR=4THENPRINTCHR$(7);M$;" TRANSACTION FILE
    EMPTY ":CHAIN"MENU.BAS"
630 PRINT:PRINT
640 IFERR<>8THENONERRORGOTO
650 PRINT"HIT ANY KEY TO RETURN ? ";:F$=INCH$(0)
660 CLOSE1:RESUME 180
```

```
0 REM TEDIT.BAS
20 ON ERROR GOTO 730
30 CL$=CHR$(27)+"E"
40 W=60
50 PRINTCL$
60 PRINTTAB(W/2-13);"Edit Transaction on file"
70 PRINT:PRINT
80 OPENOLD"1.YEAR"AS1
90 GET#1,RECORD1
100 FIELD#1,2ASTN$,2ASTY$,3ASTM$,3ASGL$
110 GL$=GL$
120 M$=TM$
130 CLOSE 1
140 OPENOLD"1."+GL$+".GL" AS I
150 GET#1,RECORD1
160 FIELD#1,2ASZ$
170 X=CVT$%(Z$)
180 DIMM%(X),A$(X)
190 GET#I,RECORD1:GOTO210
200 GET#1
210 FORS%=0TO7
220 FIELD#1,S%*30ASZ$,2ASN$,20ASA$
230 IFCVT$%(N$)<1000RA$=""THEN270
240 I%=I%+1
250 M%(I%)=CVT$%(N$):A$(I%)=A$
260 IFI%=XTHEN290
270 NEXTS%
280 GOTO200
290 CLOSE1
300 PRINT"Do you want to EDIT ";M$;" trans file
    (Y/N)? ";:AN$=INCH$(0):PRINT
```

USE: DISKFIX9 <drive > or P DISKFIX9 <drive >
Notes: * DISKFIX9 Uses 2 bytes per sector for a map table plus 640 bytes for FCB's in lower memory. DISKFIX9 runs in the TCA. If there is insufficient memory message is given DO NOT CONSOLIDATE the free chain. The sector tests will still work except for those sectors outside of the map.
* DISKFIX9 defines the free chain has any sectors that are not found in any file as free. The differences between that count and that given in SIR are reported. If there are any file collisions, truncated files, or deleted files to be recovered the free chain should not be consolidated. Delete any bad files and undelete any files to be recovered before the consolidating.
* In the near future I will have a universal version of disk drivers to operate the Tallgrass Winchester in any FLEX09 system.
* Due to the peculiarity of FLEX a bad sector ID crc error will will be reported as "drive not ready" and a data sector crc error will be reported as "read error". DISKFIX9 report not only the failing sector be also the previous sector for tracing the problem. There is a runtime prompt to have every sector/track number reported as it is read.

```
                    NAM   * DISKFIX9  03/16/82
                    ********************************************************
                    *
                    * WILLIAM HARTMANN
                    * RR2 BOX 121-L
                    * BLUERINGS, MO 64015
                    *
                    * THIS IS A REWRITE OF DISKFIX BY GEOFFREY PASS
                    *
                    * IT HAS BEEN MODIFIED FOR FLEX 6809, IT WILL HANDLE
                    * ANY TYPE OF TYPE OF DISK (HARD, S/D SINGLE/DOUBLE
                    * SIDE, SINGLE, DOUBLE DENSITY).
                    *
                    * IT DOES NOT LOCKOUT ANY SECTORS BASED ON PROTECTED FILES
                    * AS DOES THE THE 6800 VERSION
                    *
                    * IT IS CALLED AS: P DISKFIX9 (DRV0)
                    *              OR: DISKFIX9 (DRV0)
                    * IN EITHER CASE ALL PROMPTS ARE TO THE TERMINAL
                    *
                    * THIS PROGRAM TRACES ALL DIRECTORY ENTRYS AND FILES
                    * TO VERIFY THE PROPER LINKAGES AND IT CHECK FOR
                    * LINKS THAT INTERSECT
                    *
                    * THE REPORT OF THE FREE CHAIN IS DEFINED AS THOSE
                    * SECTORS NOT ASSIGNED TO ANY FILE.  IF WHERE ANY FLAWED
                    * SECTORS REMOVED BY NEWDISK THEN THE COUNT OF FREE SECTORS
                    * WILL BE MORE THAN THE NUMBER IN THE SYSTEN INFORMATION
                    * RECORD.  IF THE OPTION TO REORGANIZE THE FREE CHAIN IS USED
                    * THAN ANY FLAWED SECTORS ARE REMOVED AND THE S.I.R. IS
                    * UPDATED WITH THE CURRENT VALUES

                    * FLEX EQUATES
CC22                OSWTCH  EQU   $CC22     OUTPUT SWITCH (PRINTER)
CC20                MEMEND  EQU   $CC20     MEMORY END
CD15                GETCHR  EQU   $CD15
CD03                WARMS   EQU   $CD03     FLEX RETURN
CD18                PUTCHR  EQU   $CD18
CD1B                INBUFF  EQU   $CD1B
CD1E                PSTRNG  EQU   $CD1E     PRINT STRING W/CR-LF
CD24                PCRLF   EQU   $CD24
CD39                FL1DEC  EQU   $CD39
CD3C                OUTHEX  EQU   $CD3C
CD3F                RPTERR  EQU   $CD3F
CD45                OUTADR  EQU   $CD45
CD42                GETHEX  EQU   $CD42
9406                FMSCAL  EQU   $9406     DISK CALL

                    * SYSTEM FCB
C840                SYSFCB  EQU   $C840     SYSTEM FCB
0000                        ORG   $0
0000                DIRFCB  RMB   $140      DIRECTORY FCB
0140                SIRFCB  RMB   $140      SYSTEN INFORMATION RBC
0280                MAP     EQU   *         START OF MAP AREA

C100                        ORG   $C100
C100 20 13          START   BRA   START1

C102 04                     FCB   10         VERION 1.0

C103                MAXTRK  RMB   2          MAX TRACK/SECTOR
C105                FST_TS  EQU   *          FIRST FREE TRACK/SEC
C105                RECORD  RMB   2          RECORD COUNT
C107                RFLAG   RMB   1          <>0 RANDOM FILE, ELSE SEQU.
C108                MAPTOP  RMB   2          TOP OF SECTOR MAP
C10A                SECPRT  RMB   1          <>0 PRINT EACH SECTOR
C10B                END_TS  RMB   2          ENDING TRACK/SECTOR
C10D                PRV_TS  RMB   2          PREVIOUS TRACK/SECTOR
C10F                TRKSEC  RMB   2          CURRENT TRACK/SECTOR
C111                DIRNUM  RMB   2          DIRECTORY NUMBER
C113                SECNT   RMB   2          SECTON COUNT

C115 BD C042        START1  JSR   GETHEX     GET DRIVE #
C118 5D                     TSTB             CK FOR 0
C119 26 06                  BNE   ST20       YES
C11B 8E C51E        ST10    LDX   #IDRV
C11E 7E C505                JMP   FERR       FATAL ERROR

C121 1F 10          ST20    TFR   X,D        GET DRIVE NUM
C123 1083 0003              CMPD  #3         CK FOR VALID RANGE
C127 22 F2                  BHI   ST10       NO

                    * SET DRIVE IN FCB'S
C129 8E C840                LDX   #WRKFCB
C12C E7 03                  STB   3,X
C12E 8E 0000                LDX   #DIRFCB
C131 E7 03                  STB   3,X
C133 8E 0140                LDX   #SIRFCB
C136 E7 03                  STB   3,X

C138 CC 0003                LDD   #03        POINT TO SIR
C13B 09 C311                JSR   READ1      GET S.I.R.
C13E 27 09                  BEQ   ST40       READ OK
C140 BD CD3F                JSR   RPTERR     GIVE ERROR
C143 8E C528                LDX   #BDSIR     GIVE BAD SIR
C146 7E C505                JMP   FERR       REPORT FATAL ERROR

                    * COMPUTE MEMORY REQUIRE
                    * REQUIRES 2 BYTES FOR EACH SECTOR A NUMBER
                    * IS STORED REPRESENTING THE DIRECTORY NUMBER
                    * 0FFFF IS USED FOR THE DIRECTORY FILES

C149 BD C41B        ST40    JSR   PSIR      PRINT SIR
C14C 8E C539                LDX   #TS_MSG   T/S OK?
C14F BD C508                JSR   PROMPT
C152 27 03                  BEQ   ST60      YES
C154 7E CD03                JMP   WARMS

C157 8E 0140        ST60    LDX   #SIRFCB
C15A EC 88 66               LDD   $40+30,X  GET MAX TRACK/SECTOR
C15D FD C103                STD   MAXTRK

                    * COMPUTE MEMORY REQUIRED
C160 4C                     INCA             CONV TRACK 0-X TO 1-X+1
C161 3D                     MUL
C162 58                     ASLB             2 BYTES PER SECTOR
C163 49                     ROLA
C164 25 9E                  BCS   ST80       OVERFLOW
C166 C3 027E                ADDD  #MAP-2     OFSET START
C169 25 09                  BCS   ST80       OVERFLOW
C16B FD C108                STD   MAPTOP     SET MEMORY TOP
C16E 1083 CC20              CMPD  MEMEND     ENOUGH ROOM
C172 25 06                  BLO   ST100      YES

                    * FLAG SMALL MEMORY
C174 8E C548        ST80    LDX   #OVERMSG   PRINT WARNING
C177 BD CD1E                JSR   PSTRNG

                    * CLEAR MAP MEMORY
C17A 8E 0280        ST100   LDX   #MAP

C17D 6F 80          ST120   CLR   0,X+
C17F 8C CC20                CMPX  MEMEND
C182 23 F9                  BLS   ST120

                    * CK TO SEE IF EACH SECTOR ID TO BE PRINTED
C184 7F C10A                CLR   SECPRT     ASSUME NO
```

```
C187 8E  C593          LDX   #SECMSG
C18A BD  C508          JSR   PROMPT
C18D 26  03            BNE   ST160
C18F 7C  C10A          INC   SECPRT    YES, SET FLAG

         C192  ST160   EQU   *
               * TRACE DIRECTORY
C192 CC  0005          LDD   #$0005    FIRST TRACK/SECTOR OF DIRECTORY
C195 FD  C10F          STD   TRKSEC
C198 8E  C638          LDX   #MSG18
C19B BD  CD1E          JSR   PSTRNG
C19E BD  C31B          JSR   TRACE

               * MAIN LOOP READS EACH ENTRY IN DIRECTORY
               * AND THEN TRACES EACH ENTRY IN MAP

C1A1 CC  0000          LDD   #0
C1A4 FD  C111          STD   DIRNUM    RESET DIRECTORY NUMBER
C1A7 8E  C667          LDX   #DIRHDR   PRINT DIRECTORY HEADER
C1AA BD  CD1E          JSR   PSTRNG
C1AD 8E  0000          LDX   #DIRFCB
C1B0 86  06            LDA   #6
C1B2 A7  84            STA   0,X       OPEN DIRECTORY
C1B4 BD  0406          JSR   FMSCAL
C1B7 27  09            BEQ   DIRLOP    DIR OK

C1B9 BD  CD3F  DIRERR  JSR   RPTERR    REPORT ERROR #
C1BC 8E  C50F          LDX   #FDIR
C1BF 7E  C505          JMP   FERR

C1C2 FC  C111  DIRLOP  LDD   DIRNUM
C1C5 C3  0001          ADDD  #1
C1C8 FD  C111          STD   DIRNUM

C1CB 8E  0000          LDX   #DIRFCB
C1CE 86  07            LDA   #7
C1D0 A7  84            STA   ,X
C1D2 BD  0406          JSR   FMSCAL    GET NEXT DIRECTORY ENTRY
C1D5 26  E2            BNE   DIRERR

C1D7 30  04            LEAX  4,X       POINT TO DIRECTORY ENTRY
C1D9 A6  84            LDA   ,X        CK FIRST CHAR IN NAME
C1DB 27  2C            BEQ   DIREND    LAST ENTRY
C1DD BD  C42C          JSR   PRTNAM    PRINT DIRECTORY NAME
C1E0 8E  0004          LDX   #DIRFCB+4
C1E3 A6  84            LDA   ,X        CK FOR DELETED FILE
C1E5 2B  BB            BMI   DIRLOP    YES

C1E7 EC  00            LDD   13,X      GET STARTING TRACK SECTOR
C1E9 FD  C10F          STD   TRKSEC
C1EC EC  0F            LDD   15,X      GET ENDING TRACK SECTOR
C1EE FD  C10B          STD   END_TS
C1F1 EC  88 11         LDD   17,X
C1F4 FD  C113          STD   SECNT     SET SECTOR COUNT
C1F7 7F  C107          CLR   RFLAG     ASSUME SEQUENCE FILE
C1FA A6  88 13         LDA   19,X
C1FD 27  05            BEQ   DIR20
C1FF 86  02            LDA   #2        SET RANDOM FLAG
C201 B7  C107          STA   RFLAG
C204 BD  C31B  DIR20   JSR   TRACE     TRACE FILE
C207 20  B9            BRA   DIRLOP    DO NEXT

         C209  DIREND  EQU   *

               * COUNT NUMBER OF FREE SECTORS
               * FIND START OF 1ST TRACK IN MAP
C209 4F                CLRA
C20A F6  C104          LDB   MAXTRK+1  SECTOR SIZE
C20D 58                ASLB
C20E 49                ROLA            * 2
C20F C3  0280          ADDD  #MAP
C212 34  06            PSHS  D         SAVE MAP START FOR LATER
C214 1F  01            TFR   D,X
C216 108E 0000         LDY   #0        COUNTER

C21A EC  81    FC10    LDD   0,X++     CK FOR ZERO
C21C 26  02            BNE   FC20      NO, SECTOR IN USE
C21E 31  21            LEAY  1,Y
C220 8C  C10B  FC20    CMPX  #MAPTOP
C223 23  F5            BLS   FC10
```

```
C225 10BF C113         STY   SECNT     SET COUNT
C229 8E  C5A8          LDX   #FCMSG    GIVE FREE CHAIN COUNT
C22C BD  CD1E          JSR   PSTRNG
C22F 8E  C113          LDX   #SECNT    # IN MAP
C232 BD  C4E0          JSR   BDCS
C235 8E  01A1          LDX   #SIRFCB+640+33 GET F.C. COUNT IN SIR
C238 BD  C4E0          JSR   BECS
               * CK FOR FREE CHAIN CORRECTION/DATE/ION
C23B 8E  C5C3          LDX   #FCPMNT
C23E BD  C508          JSR   PROMPT
C241 27  03            BEQ   FC100     YES
C243 7E  CD03          JMP   WARMS     NO, DONE

C246 35  20    FC100   PULS  Y         SET MAP START
C248 CC  0000          LDD   #0
C24B 1F  03            TFR   D,U       RESET SECTOR COUNT
C24D FD  C105          STD   FST_TS    RESET 1ST FREE TS
C250 FD  C10B          STD   END_TS
C253 FD  C10D          STD   PRV_TS
               * SET TO 1ST TR/SEC-1
C256 CC  0100          LDD   #$0100
C259 BD  C2FA          JSR   FNDFRE    GET NEXT FREE SECTOR
C25C FD  C10F          STD   TRKSEC    SET CURRENT

C25F BD  C2FA  FC200   JSR   FNDFRE    GET NEXT FREE SECTOR
C262 FD  C10B          STD   END_TS
C265 8E  C840  FC205   LDX   #MARKFCB
C268 FC  C10F          LDD   TRKSEC
C26B BD  C311          JSR   READ1     GET 1 SECTOR
C26E 26  3C            BNE   FC300     BAD SECTOR
C270 FC  C10B          LDD   END_TS    GET LINK
C273 ED  88 40         STD   $40,X     SET LINK
C276 86  0A            LDA   #10
C278 A7  84            STA   ,X        REWRITE SECTOR
C27A BD  0406          JSR   FMSCAL
C27D 26  4D            BNE   FC300     BAD WRITE

C27F 7D  C10A          TST   SECPRT    PRINT ALL
C282 27  06            BEQ   FC210     NO
C284 8E  C10F          LDX   #TRKSEC
C287 B9  C4F4          JSR   PRTSEC

C28A 70  C105  FC210   TST   FST_TS    HAS FREE CHAIN START BEEN SET
C28D 26  06            BNE   FC220     YES
C28F FC  C10F          LDD   TRKSEC
C292 FD  C105          STD   FST_TS    SET START
C295 33  41    FC220   LEAU  1,U       COUNT SECTOR
C297 FC  C10F          LDD   TRKSEC    UPDATE PREVIOUS
C29A FD  C10D          STD   PRV_TS
C29D FC  C10B          LDD   END_TS    UPDATE CURRENT
C2A0 FD  C10F          STD   TRKSEC
C2A3 26  BA            BNE   FC200     DO AGAIN

               * NEXT=0 END LINK, UPDATE SIR
C2A5 8E  0140          LDX   #SIRFCB
C2A8 FC  C105          LDD   FST_TS
C2AB ED  88 5D         STD   $40+29,X  SET START OF FREE CHAIN
C2AE FC  C10B          LDD   PRV_TS    SET END OF FREE
C2B1 ED  88 5F         STD   $40+31,X
C2B4 EF  88 61         STU   $40+33,X  SET FREE COUNT
C2B7 86  0A            LDA   #10
C2B9 A7  84            STA   ,X        REWRITE SIR
C2BB BD  0406          JSR   FMSCAL
C2BE 27  06            BEQ   FC240     OK
C2C0 8E  C520          LDX   #BSIR
C2C3 7E  C505          JMP   FERR

               * OUTPUT CURRENT CURRENT SIR
C2C6 BD  C41D  FC240   JSR   PSIR
C2C9 7E  CD03          JMP   WARMS

               * BAD SECTOR FIX LINK
C2CC BD  CD3F  FC300   JSR   RPTERR    DISK ERROR
C2CF 8E  C605          LDX   #SECN
C2D2 BD  C407          JSR   TRCERR

C2D5 70  C105          TST   FST_TS    FIRST FREE SEC
C2D8 27  17            BEQ   FC340     YES
C2DA FC  C10D          LDD   PRV_TS    BACK UP
C2DD 1093 C10F         CMPD  TRKSEC    ALREADY BACKED UP
C2E1 26  06            BNE   FC320
```

```
410 PRINT:PRINT
420 PRINT"Enter END to return to MENU"
430 INPUT"Enter ACCT@ Debited ....",AN$
440 IFAN$="END"THENB80ELSED%=VAL(AN$)
450 FORJ%=1TO%:IFD%=N%(J%)THEN470ELSENEXTJ%
460 PRINTCHR$(7);"ACCT@";D%;"NOT FOUND (REENTER)"
    :GOTO430
470 D$=A$(J%)
480 INPUT"Enter ACCT@ Credited ...",AN$
490 IFAN$="END"THENB80ELSEC%=VAL(AN$)
500 FORJ%=1TO%:IFC%=N%(J%)THEN520ELSENEXTJ%
510 PRINTCHR$(7);"ACCT@";C%;"NOT FOUND (REENTER)"
    :GOTO480
520 C$=A$(J%)
530 IFD%=C%THENPRINTCHR$(7);"DEBIT AND CREDIT ARE
    SAME ACCT@ NUMBER":GOTO410
540 INPUT"Enter Check Number .....",CN%
550 INPUT"Enter PAYEE/SOURCE.....",P$
560 P$=LEFT$(P$,26)
570 INPUT"Enter Amount ...........",A
580 IFY$=""THEN640
590 PRINT"Date (Return if correct)= ";Y$
600 PRINT"ELSE Enter NEW Month ...";
610 INPUTLINEAN$
620 IFAN$=""THEN690
630 IFVAL(AN$)<>0THEN590ELSEAN$=LEFT$(AN$,3)
640 INPUT"Enter DAY (1-31) .......",AD$
650 IFVAL(AD$)<1 OR VAL(AD$)>31THEN640
660 IFLEN(AD$)<2THENAD$="0"+AD$
670 IFY$=""THENY$=M$+"/"+AD$:GOTO590
680 Y$=AN$+"/"+AD$
690 PRINTCL$
700 IFD%/100=1 OR D%/100=4 THEND$="+"+D$ ELSE
    D$="-"+D$
710 IFC%/100=1 OR C%/100=4 THENC$="-"+C$ ELSE
    C$="+"+C$
720 PRINT:PRINT:PRINT"DEBIT> ";D$;TAB(30);"CREDIT> ";
    C$
730 PRINT:PRINT
740 PRINT"DATE OF TRANSACTION IS ";Y$
750 PRINTUSING'CHECK ######  TO
    \2345678901234567890123456\ FOR $$$,$$$.$$',
    CN%,P$,A
760 PRINT:PRINT"IS THIS CORRECT DATA (Y/N)? ";
    :F$=INCH$(0)
770 IFF$="N"THEN370
780 IFF$<>"Y"THEN760
790 T%=T%+1:R%=T%/5:S%=T%-(R%*5)
800 IFS%=0THENFIELD#1,252AS%$:LSETT%$=""
    :PUT#1,RECORDR%+1
810 GET#1,RECORDR%+1
820 FIELD#1,S%*50AS%$,2ASTD$,2ASTC$,2ASTN$,
    10ASTY$,26ASTP$,8ASTA$
830 LSETTD$=CVT%$(D%):LSETTC$=CVT%$(C%)
    :LSETTN$=CVT%$(CN%)
840 LSETTY$=Y$:LSETTP$=P$:LSETTA$=CVTF$(A)
850 PUT#1,RECORDR%+1
860 GOSUB890
870 GOTO 370
880 CLOSE1:CHAIN"MENU.BAS"
890 REM ADD TO FILE SIZE
900 GET#1,RECORD1:FIELD#1,2AS%$
910 LSET%$=CVT%$(T%)
920 PUT#1,RECORD1
930 RETURN
940 IFERR=4THENCLOSE1:OPENNEW"1."+M$AS1
    :FIELD#1,2AS%$:LSET%$=CVT%$(0):PUT#1:RESUME350
950 IFERR=3OTHENPRINT"DATA MISMATCH REENTER":RESUME
960 ON ERROR GOTO
```

TO BE CONTINUED

ARCADE-50

Hal W. Snyder
855 Fairoaks
Deerfield, IL 60015

**** ARCADE-50 BOARD: DESCRIPTION AND OS-9 DRIVER ****

The ARCADE-50 board made by Terminus Design of Ellenwood, GA. gives the user substantial capability for color graphics, sound generation, and analog input. The report I am including below mentions what I have found to be some strengths and weaknesses of the board, compares its TMS9918A video chip with Motorola's 6847 (used in Tandy's Color Computer), summaries hardware control protocols (some of which were not readily discernible from documentation provided), and concludes with listings of (1) general-purpose 6809 assembly driver and (2) BASIC09 exerciser.

I. Strengths & Weaknesses.

Strengths are the hardware capabilities of the board:

1. TMS 9918A video chip (discussed below) with 16K of on-board dynamic RAM. You can do overlay of external analog video for subtitling, superimposed images, etc.; so, this system will allow the sorts of interactive videodisc systems discussed in the June '82 BYTE magaine.
2. 8-channel x 8-bit ADC (10,000 samples/sec - chip is ADC0809).
3. Three (3) AY-3-8910 sound chips (PSG's). Each of these has three tones & white noise with programmable pitch, amplitude, and envelope - but only one envelope at a time per chip - along with two 8-bit parallel I/O ports. There are two audio amplifiers on board, with jumper selection of stereo (two PSG's to one channel and the third PSG to the other channel) or mono (all three PSG's to both channels). Each port is either all 8 bits in or all 8 out, but not all ports are readily accessible to you (see complaints).

Complaints: with all that hardware, there is still plenty of real estate on the board, yet some corners seem to have been cut unnecessarily:

1. Only address lines A15-A8 and A2-A0 are decoded. This means that although the board requires only six bytes of memory space, it occupies 256. Also, it means that extended SS-50C addressing is not supported outside the first 64K.
2. Of the parallel ports on board only PSG1's B port and half of PSG2's B port go to external connectors. The other four and a half ports are fully programmable, but remain simply as floating pins on the chips they originate from. So, if you want to use these latter ports, you will have to solder or piggyback.
3. There is no on-board RF modulator. This means that you must supply either modulator plus conventional TV or else NTSC video monitor to see the video output. BR 4. Although all essential connectors and cables are made available by Terminus Design, you will need to provide mounting for & connect cables and offboard connectors yourself. This is a minor soldering job.
5. Documentation is limited. While component layout and functional block diagrams are provided, there is no schematic diagram. Support software for 6809 is in the manual & on two floppies. Source listings illustrating some but not all hardware functions are provided on paper, while additional functions are demonstrated by programs provided in source form on diskette, but nowhere is there a complete explanation of driver protocol for all functions. Included is a PAC-MAN type game which I could not get to work using my GIMIX 80x24 based console terminal. The game works properly, and is quite impressive, if system console is a serial port

at $E004. Support software has been provided in machine-language-only form in something grandiosely called the Terminus Design Multitasking System. This program must have been a labor of love on someone's part but is hardly a complete operating system and does not even support position-independent code. For me, this program was of limited value except for running the demo game, especially since source was not provided.

II. Video chips.

The following comparison is adapted from TI's preliminary data sheet on

	TMS9918A	MC6847
Border colors	15	2
Text mode		
Screen	40x24	32x16
Character matrix	5x7	5x7
Character set	256	128
Character generator	RAM	ROM
Colors	15	2
Graphics at max. resolution	15 colors 256x192 (see note)	2 colors 256x192 or 4 colors 128x192
Video RAM supported	16K	4K

Note: there is a limitation in use of colors for TI's chip in that in any one of the 192 rows, in any aligned 8-dot segment of that row, only 2 colors may be used. The TMS9918A also has functions not found at all on the 6847. There is a "transparent" color for use with background analog video. There are 32 "sprites" available per screen: medium-sie figures which may be mapped to any part of the screen. There is on-chip refresh and multiplex circuitry for dynamic RAM's for the 16K of video RAM (VRAM). Composite video is generated on-chip.

I have presented this comparison because Tandy's Color Computer, also 6809-based, uses Motorola's 6847 chip. My feeling is, Color Computer, move over, at least for the roll-your-own software experimenter. (For audio/video programmers: the Color Computer is also limited because it has no intrinsic interrupt-driven I/O and only limited timer capability, so that it is very tricky to overlap any two of general program execution, sound generation, I/O, and interval timing.)

III. Driver protocols for the ARCADE-50. "Rn" refers to board register #n, accessed by read or write to board address+n. "PPI" refers to the on-board 8255A chip. "VDP" is the TMS9918A.

A. Read ADC channel #n, n=0-7.
   1. #n -> R1.
   2. $C0 -> R2.
   3. $00 -> R2.
   4. Wait about 64usec.
   5. R0 -> 8-bit data.

B. Write 8-bit data #dd to register r, r=00-0F, of one PSG.
   1. r -> R1.
   2. ss -> R2. ss=$0A for PSG0, $30 for PSG1, $05 for PSG2.
   3. $00 -> R2.
   4. #dd -> R1.
   5. tt -> R2. tt=$00 for PSG0, $10 for PSG1, $04 for PSG2.
   6. $00 -> R2.
   If SPST joystick switches are used, then when writing to PSG2, register 7, bit 7 should always be written as zero to avoid shorting a port output.

C. Read 8-bit data #dd from register r, r=00-0F, of one PSG.
   1. r -> R1.
   2. ss -> R2. ss as above.
   3. $00 -> R2.
   4. $92 -> R3. Configure PPI(B) as input.
   5. uu -> R2. uu=$02 for PSG0, $20 for PSG1, $01 for PSG2.
   6. R1 -> #dd.
   7. $00 -> R2.
   8. $90 -> R3. Back to normal. PPI(B) as output.

D. Read VDP status.
   1. Read R5.

E. Write 8-bit #dd to VDP register r, r=0-7.
   1. #dd -> R5.
   2. $8r -> R5.

F. Read #dd from VRAM address #aabb, aabb=0000-3FFF.
   1. #bb -> R5.
   2. #aa -> R5. Two highest-order bits must be zero!
   3. R5 -> #dd.

G. Store #dd into VRAM address #aabb, aabb=0000-3FFF.
   1. #bb -> R5.
   2. #aa+$40 -> R5.
   3. #dd -> R4.

H. Initalize.
   1. $90 -> R3. PPI mode 0: A in, B & C out.
   2. $00 -> R2.
   3. $07 -> R1. Steps 3-8: $7F to reg 7, PSG 2.
   4. $05 -> R2.
   5. $00 -> R2.
   6. $7F -> R1.
   7. $04 -> R2.
   8. $00 -> R2.

I. Read joystick switches.
   1. $0F -> R1. Steps 1-8: read reg $F (port B), PSG2.
   2. $05 -> R2.
   3. $00 -> R2.
   4. $92 -> R3.
   5. $01 -> R2.
   6. R1 -> data in low 4 bits. 0=button pushed.
   7. $00 -> R2.
   8. $90 -> R3.

IV. Programs. ARCM is a series of OS-9 assembler compatible routines to perform all of the above and some additional functions. It is callable by either BASIC09 or assembler, and is of course position-independent. ARBUG is a BASIC09 program using ARCM. It allows the user to hand- program the ARCADE-50. It includes a "$" command allowing escape to OS-9, so that software developers may use it in combination with another ARCADE-50 program being debugged. If anyone is interested, I can make these programs available via VIDEOTEX. I also have a full upper- and lower-case ASCII character generator file for the text mode.

DISKFIX9

William Hartmann
RR 2 Box 121-1
Blue Springs, Mo 64015

April 12, 1982

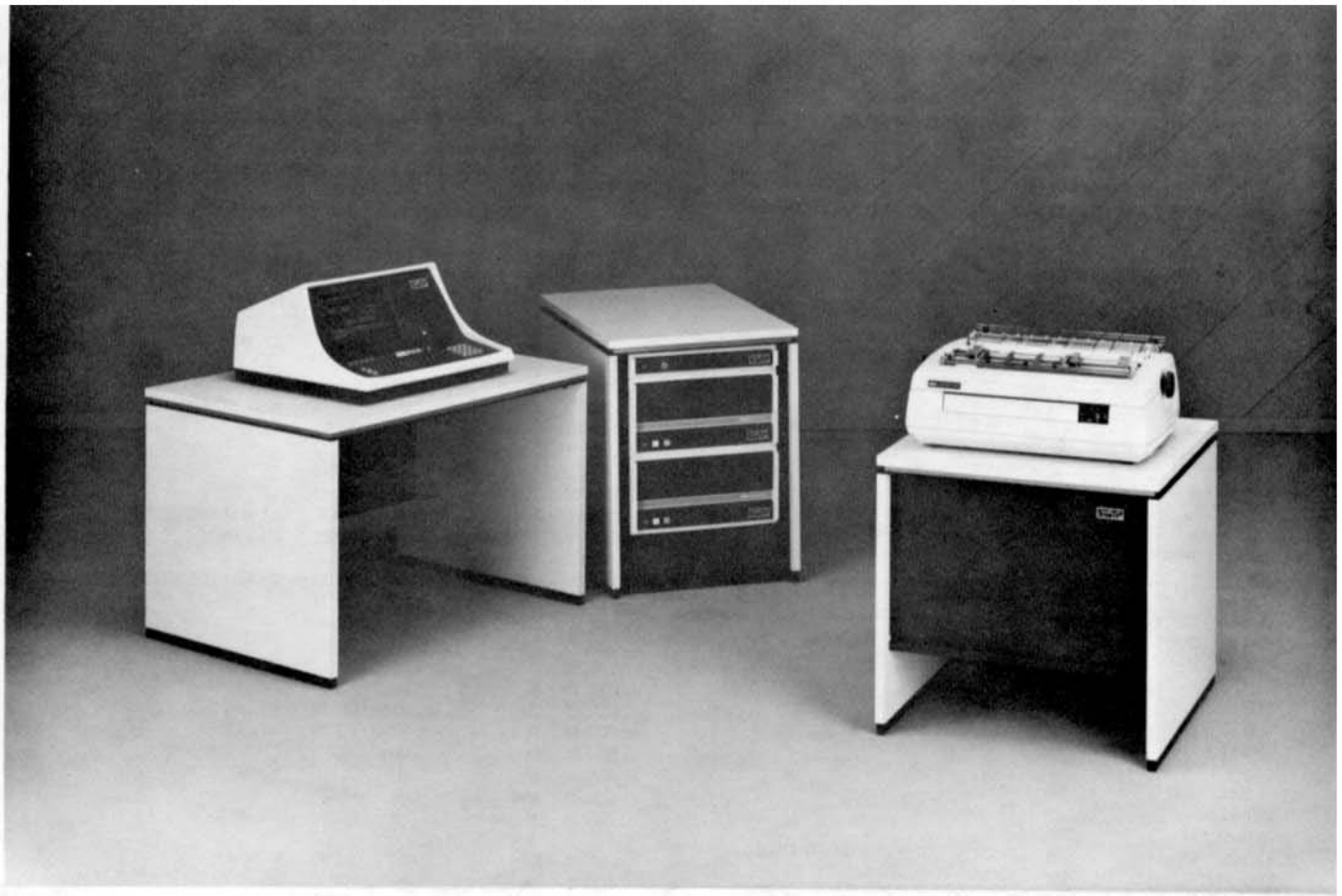
I have a 6809 system with 5 inch, double density, double sided floppy and Tallgrass 5 inch Winchester. I am using the General version of FLEX so that I could write the disk drivers for my unique system. In developing my system I used the TSC disk diagnostic to find and correct my problems. However, VALIDATE will only work with "standard" size disks.

I started out to modify the DISKFIX program (by Geoffrey Pass, Dec, 1981 MICRO Journal) and ended up writting a new program DISKFIX9. DISKFIX9 reads each sector of the directory and of the data files. Each sector read is verified for collision with other files, the proper record number and the correct ending sector number. DISKFIX9 optionally will consolidate the free chain.

USE: DISKFIX9 <drive > or P DISKFIX9 <drive >
Notes: * DISKFIX9 Uses 2 bytes per sector for a map table plus 640 bytes for FCB's in lower memory. DISKFIX9 runs in the TCA. If there is insufficient memory message is given DO NOT CONSOLIDATE the free chain. The sector tests will still work except for those sectors outside of the map.
* DISKFIX9 defines the free chain has any sectors that are not found in any file as free. The differences between that count and that given in SIR are reported. If there are any file collisions, truncated files, or deleted files to be recovered the free chain should not be consolidated. Delete any bad files and undelete any files to be recovered before the consolidating.
* In the near future I will have a universal version of disk drivers to operate the Tallgrass Winchester in any FLEX09 system.
* Due to the peculiarity of FLEX a bad sector ID crc error will will be reported as "drive not ready" and a data sector crc error will be reported as "read error". DISKFIX9 report not only the failing sector be also the previous sector for tracing the problem. There is a runtime prompt to have every sector/track number reported as it is read.

```
                    NAM   * DISKFIX9  03/16/82
                    ********************************************************
                    *
                    * WILLIAM HARTMANN
                    * 892 BOI 121-1
                    * BLUEARTHS, MO 64015
                    *
                    * THIS IS A REWRITE OF DISKFIX BY GEOFFREY PASS
                    *
                    * IT WAS BEEN MODIFIED FOR FLEX 6809, IT WILL HANDLE
                    * ANY TYPE OF TYPE OF DISK (HARD, 5/8 SINGLE/DOUBLE
                    * SIDE, SIGNLE, DOUBLE DENSTIY).
                    *
                    * IT DOES NOT LOCKOUT ANY SECTORS BASED ON PROTECTED FILES
                    * AS DOES THE THE 6800 VERSION
                    *
                    * IT IS CALLED AS: P DISKFIX9 <DRV0>
                    *            OR: DISKFIX9 <DRV0>
                    * IN EITHER CASE ALL PROMPTS ARE TO THE TERMINAL
                    *
                    * THIS PROGRAM TRACES ALL DIRECTORY ENTRYS AND FILES
                    * TO VERIFY THE PROPER LINKAGES AND IT CHECK FOR
                    * LINKS THAT INTERSECT
                    *
                    * THE REPORT OF THE FREE CHAIN IS DEFINED AS THOSE
                    * SECTORS NOT ASSIGNED TO ANY FILE.  IF WHERE ANY FLAWED
                    * SECTORS REMOVED BY MENDISK THEN THE COUNT OF FREE SECTORS
                    * WILL BE MORE THAN THE NUMBER IN THE SYSTEM INFORMATION
                    * RECORD.  IF THE OPTION TO REORGANIZE THE FREE CHAIN IS USED
                    * THAN ANY FLAWED SECTORS ARE REMOVED AND THE S.I.R. IS
                    * UPDATED WITH THE CURRENT VALUES

                    * FLEX EQUATES
CC22                DSWTCH  EQU   $CC22    OUTPUT SWITCH (PRINTER)
CC28                MEMEND  EQU   $CC28    MEMORY END
CD15                GETCHR  EQU   $CD15
C003                WARMS   EQU   $C003    FLEX RETURN
CD18                PUTCHR  EQU   $CD18
CD1B                INBUFF  EQU   $CD1B
CD1E                PSTRNG  EQU   $CD1E    PRINT STRING W/CR-LF
CD24                PCRLF   EQU   $CD24
CD39                FLIDEC  EQU   $CD39
CD3C                OUTHEX  EQU   $CD3C
CD3F                RPTERR  EQU   $CD3F
CD45                OUTADR  EQU   $CD45
CD42                GETNEX  EQU   $CD42
0406                FMSCAL  EQU   $0406    DISK CALL

C840                SYSFCB  EQU   $C840    SYSTEM FCB
0000                        ORG   $0
0000                DIRFCB  RMB   $140     DIRECTORY FCB
0140                SIRFCB  RMB   $140     SYSTEM INFORMATION REC
0280                MAP     EQU   *        START OF MAP AREA

C100                        ORG   $C100
C100 20  13         START   BRA   START1

C102 0A                     FCB   10      VERION 1.0

C103                MAXTRK  RMB   2       MAX TRACK/SECTOR
     C105           FST_TS  EQU   *       FIRST FREE TRACK/SEC
C105                RECORD  RMB   2       RECORD COUNT
C107                RFLAG   RMB   1       <>0 RANDOM FILE, ELSE SEQU.
C108                MAPTOP  RMB   2       TOP OF SECTOR MAP
C10A                SECPRT  RMB   1       <>0 PRINT EACH SECTOR
C10B                END_TS  RMB   2       ENDING TRACK/SECTOR
C10D                PRV_TS  RMB   2       PREVIOUS TRACK/SECTOR
C10F                TRKSEC  RMB   2       CURRENT TRACK/SECTOR
C111                DIRNUM  RMB   2       DIRECTORY NUMBER
C113                SECNT   RMB   2       SECTOR COUNT

C115 B0  C042       START1  JSR   GETNEX   GET DRIVE 0
C118 50                     TSTB          CK FOR 0
C119 26  06                 BNE   ST20     YES
C11B 8E  C51E       ST10    LDX   #IDRV
C11E 7E  C505               JMP   FERR    FATAL ERROR

C121 1F  10         ST20    TFR   X,D      GET DRIVE NUM
C123 1083 0003              CMPD  #3      CK FOR VALID RANGE
C127 22  F2                 BHI   ST10    NO

                    * SET DRIVE IN FCB'S
C129 8E  C840               LDX   #MAKFCB
C12C E7  03                 STB   3,X
C12E 8E  0000               LDX   #DIRFCB
C131 E7  03                 STB   3,X
C133 8E  0140               LDX   #SIRFCB
C136 E7  03                 STB   3,X

C138 CC  0003               LDD   #03     POINT TO SIR
C13B BD  C311               JSR   READ1   GET S.I.R.
C13E 27  09                 BEQ   ST40    READ OK
C140 BD  CD3F               JSR   RPTERR  GIVE ERROR
C143 8E  C528               LDX   #BDSIR  GIVE BAD SIR
C146 7E  C505               JMP   FERR    REPORT FATAL ERROR

                    * COMPUTE MEMORY REQUIRE
                    * REQUIRES 2 BYTES FOR EACH SECTOR A NUMBER
                    * IS STORED REPRESENTING THE DIRECTORY NUMBER
                    * $FFFF IS USED FOR THE DIRECTORY FILES

C149 BD  C41D       ST40    JSR   PSIR    PRINT SIR
C14C 8E  C539               LDX   #TS_MSG T/S OK?
C14F BD  C50B               JSR   PROMPT
C152 27  03                 BEQ   ST60    YES
C154 7E  CD03               JMP   WARMS

C157 8E  0140       ST60    LDX   #SIRFCB
C15A EC  88 66              LDD   $40+3B,X GET MAX TRACK/SECTOR
C15D FD  C103               STD   MAXTRK

                    * COMPUTE MEMORY REQUIRED
C160 4C                     INCA          CONV TRACK 0-X TO 1-X+1
C161 3D                     MUL
C162 58                     ASLB          2 BYTES PER SECTOR
C163 49                     ROLA
C164 25  0E                 BCS   ST80    OVERFLOW
C166 C3  027E               ADDD  #MAP-2  OFSET START
C169 25  09                 BCS   ST80    OVERFLOW
C16B FD  C108               STD   MAPTOP  SET MEMORY TOP
C16E 1083 CC28              CMPD  MEMEND  ENOUGH ROOM
C172 25  06                 BLO   ST100   YES

                    * FLAG SMALL MEMORY
C174 8E  C54D       ST80    LDX   #MEMMSG PRINT WARNING
C177 BD  CD1E               JSR   PSTRNG

                    * CLEAR MAP MEMORY
C17A 8E  0280       ST100   LDX   #MAP

C17D 6F  80         ST120   CLR   0,X+
C17F 8C  CC28               CMPX  #MEMEND
C182 23  F9                 BLS   ST120

                    * CK TO SEE IF EACH SECTOR IS TO BE PRINTED
C184 7F  C10A               CLR   SECPRT  ASSUME NO
```

```
C187 8E  C593          LDX   #SECMDE
C18A BD  C508          JSR   PROMPT
C18D 26  03            BNE   ST160
C18F 7C  C10A          INC   SECPRT    YES, SET FLAG

         C192  ST160   EQU   *
               * TRACE DIRECTORY
C192 CC  0005          LDD   #$00005   FIRST TRACK/SECTOR OF DIRECTORY
C195 FD  C10F          STD   TRKSEC
C198 8E  C630          LDX   #DIR
C19B BD  CD1E          JSR   PSTRNG
C19E BD  C318          JSR   TRACE

               * MAIN LOOP READS EACH ENTRY IN DIRECTORY
               * AND THEN TRACES EACH ENTRY IN MAP

C1A1 CC  0000          LDD   #0
C1A4 FD  C111          STD   DIRNUM    RESET DIRECTORY NUMBER
C1A7 8E  C667          LDX   #DIRHDR   PRINT DIRECTORY HEADER
C1AA BD  CD1E          JSR   PSTRNG
C1AD 8E  0000          LDX   #DIRFCB
C1B0 86  06            LDA   #6
C1B2 A7  84            STA   0,X       OPEN DIRECTORY
C1B4 BD  D406          JSR   FMSCAL
C1B7 27  09            BEQ   DIRLOP    DIR OK

C1B9 BD  CD3F  DIRERR  JSR   RPTERR    REPORT ERROR #
C1BC 8E  C50F          LDX   #FDIR
C1BF 7E  C505          JMP   FERR

C1C2 FC  C111  DIRLOP  LDD   DIRNUM
C1C5 C3  0001          ADDD  #1
C1C8 FD  C111          STD   DIRNUM

C1CB 8E  0000          LDX   #DIRFCB
C1CE 86  07            LDA   #7
C1D0 A7  84            STA   ,X
C1D2 BD  D406          JSR   FMSCAL    GET NEXT DIRECTORY ENTRY
C1D5 26  E2            BNE   DIRERR

C1D7 30  04            LEAX  4,X       POINT TO DIRECTORY ENTRY
C1D9 6D  84            TST   ,X        CK FIRST CHAR IN NAME
C1DB 27  2C            BEQ   $1REND    LAST ENTRY
C1DD BD  C42C          JSR   PRTNAM    PRINT DIRECTORY NAME
C1E0 8E  0004          LDX   #DIRFCB+4
C1E3 6D  84            TST   ,X        CK FOR DELETED FILE
C1E5 2B  08            BMI   DIRLOP    YES

C1E7 EC  00            LDD   13,X      GET STARTING TRACK SECTOR
C1E9 FD  C10F          STD   TRKSEC
C1EC EC  0F            LDD   15,X      GET ENDING TRACK SECTOR
C1EE FD  C10B          STD   END_TS
C1F1 EC  88 11         LDD   17,X
C1F4 FD  C113          STD   SECNT     SET SECTOR COUNT
C1F7 7F  C107          CLR   RFLAG     ASSUME SEQUENCE FILE
C1FA 6D  88 13         TST   19,X
C1FD 27  05            BEQ   DIR20
C1FF 86  02            LDA   #2        SET RANDOM FLAG
C201 B7  C107          STA   RFLAG
C204 BD  C318  DIR20   JSR   TRACE     TRACE FILE
C207 20  B9            BRA   DIRLOP    DO NEXT

         C209  DIREND  EQU   *

               * COUNT NUMBER OF FREE SECTORS
               * FIND START OF 1ST TRACK IN MAP
C209 4F                CLRA
C20A F6  C104          LDB   MAXTRK+1  SECTOR SIZE
C20D 58                ASLB
C20E 49                ROLA            * 2
C20F C3  0280          ADDD  #MAP
C212 34  06            PSHS  D         SAVE MAP START FOR LATER
C214 1F  01            TFR   D,X
C216 108E 0000         LDY   #0        COUNTER

C21A EC  81    FC10    LDD   0,X++     CK FOR ZERO
C21C 26  02            BNE   FC20      NO, SECTOR IN USE
C21E 31  21            LEAY  1,Y
C220 8C  C10B  FC20    CMPX  #MAPTOP
C223 23  F5            BLS   FC10

C225 108F C113         STY   SECNT     SET COUNT
C229 8E  C5A8          LDX   #FCMSG    GIVE FREE CHAIN COUNT
C22C BD  CD1E          JSR   PSTRNG
C22F BE  C113          LDX   #SECNT    # IN MAP
C232 BD  C4E8          JSR   #EC5
C235 8E  01A1          LDX   #$1RFCB+$40+33 GET F.C. COUNT IN SIR
C238 BD  C4E8          JSR   #EC5
               * CK FOR FREE CHAIN CONSOLIDATEION
C23B 8E  C5C3          LDX   #FCPRMT
C23E BD  C50D          JSR   PROMPT
C241 27  03            BEQ   FC100     YES
C243 7E  CD03          JMP   WARMS     NO, DONE

C246 35  20    FC100   PULS  Y         GET MAP START
C248 CC  0000          LDD   #0
C24B 1F  03            TFR   D,U       RESET SECTOR COUNT
C24D FD  C105          STD   FST_TS    RESET 1ST FREE TS
C250 FD  C10B          STD   END_TS
C253 FD  C10D          STD   PRV_TS
               * SET TO 1ST TR/SEC-1
C256 CC  0100          LDD   #$0100
C259 BD  C2FA          JSR   FNDFRE    GET NEXT FREE SECTOR
C25C FD  C10F          STD   TRKSEC    SET CURRENT

C25F BD  C2FA  FC200   JSR   FNDFRE    GET NEXT FREE SECTOR
C262 FD  C10B          STD   END_TS
C265 8E  CB40  FC205   LDX   #WRKFCB
C268 FC  C10F          LDD   TRKSEC
C26B BD  C311          JSR   READ1     GET 1 SECTOR
C26E 26  5C            BNE   FC300     BAD SECTOR
C270 FC  C10B          LDD   END_TS    GET LINK
C273 ED  88 40         STD   $40,X     SET LINK
C276 86  0A            LDA   #10
C278 A7  84            STA   ,X        REWRITE SECTOR
C27A BD  D406          JSR   FMSCAL
C27D 26  40            BNE   FC300     BAD WRITE

C27F 70  C10A          TST   SECPRT    PRINT ALL
C282 27  06            BEQ   FC210     NO
C284 8E  C10F          LDX   #TRKSEC
C287 BD  C4F4          JSR   PRTSEC

C28A 70  C105  FC210   TST   FST_TS    HAS FREE CHAIN START BEEN SET
C28D 26  06            BNE   FC220     YES
C28F FC  C10F          LDD   TRKSEC
C292 FD  C105          STD   FST_TS    SET START
C295 33  41    FC220   LEAU  1,U       COUNT SECTOR
C297 FC  C10F          LDD   TRKSEC    UPDATE PREVIOUS
C29A FD  C10D          STD   PRV_TS
C29D FC  C10B          LDD   END_TS    UPDATE CURRENT
C2A0 FD  C10F          STD   TRKSEC
C2A3 26  BA            BNE   FC200     DO AGAIN

               * NEXT=0 END LINK, UPDATE SIR
C2A5 8E  0140          LDX   #$1RFCB
C2A8 FC  C105          LDD   FST_TS
C2AB ED  88 50         STD   $40+29,X  SET START OF FREE CHAIN
C2AE FC  C10D          LDD   PRV_TS    SET END OF FREE
C2B1 ED  88 5F         STD   $40+31,X
C2B4 EF  88 61         STU   $40+33,X  SET FREE COUNT
C2B7 86  0A            LDA   #10
C2B9 A7  84            STA   ,X        REWRITE SIR
C2BB BD  0406          JSR   FMSCAL
C2BE 27  06            BEQ   FC240     OK
C2C0 8E  C528          LDX   #MSIR
C2C3 7E  C505          JMP   FERR

               * OUTPUT CURRENT CURRENT SIR
C2C6 BD  C41D  FC240   JSR   PSIR
C2C9 7E  CD03          JMP   WARMS

               * BAD SECTOR FIX LINK
C2CC BD  CD3F  FC300   JSR   RPTERR    DRIVER ERROR
C2CF 8E  C605          LDX   #SECN
C2D2 BD  C407          JSR   TRCERR

C2D5 70  C105          TST   FST_TS    FIRST FREE SEC
C2D8 27  17            BEQ   FC340     YES
C2DA FC  C10D          LDD   PRV_TS    BACK UP
C2DD 10B3 C10F         CMPD  TRKSEC    ALREADY BACKED UP
C2E1 26  06            BNE   FC320
```



THE COMPLETE BUSINESS SYSTEM

+Multiuser+Highly Expandable+Cost Effective

S+ THE CONCEPT

The S+ system is a modular computer system in which all portions of the hardware and software are designed to work together in the most efficient way possible. An S+ single user system with floppy disk storage is a competitive and cost effective entry level system. Unlike most other small computers being sold as "personal", or "small business" machines, the S+ system may be expanded to maximum capabilities using this same hardware and software. You cannot end up with a DEAD END system that cannot be expanded and whose software is not compatible with larger machines. A basic S+ system may be expanded to thirty-two users, a megabyte of main memory and hundreds of megabytes of hard disk storage by simply plugging in, or connecting the desired upgrade equipment.

TOTAL DESIGN—Hardware and Software

The S+ system is an integrated hardware and software design. The two complement and enhance each other in this system. The UniFLEX® operating system used in the S+ systems is patterned after the Bell Laboratories UNIX® operating system, one of the most admired and widely used operating systems in the world. Instead of being an afterthought, the software is part of the design of the S+ system. You can be sure that with this approach that all parts of the computer operate with maximum efficiency and cost effectiveness.

THE CENTRAL PROCESSOR

The basic S+ system is configured with 256K bytes of memory and can be expanded to more than 1 million bytes. An efficient and fast hardware memory management system is used to allocate the available memory among the users on a dynamic basis. As little as 8K bytes, or the entire memory—if needed—can be used by any individual user. This makes it possible to run very large programs on the system, but it also uses no more memory than necessary for a particular job. The increase in cost effectiveness of this system over crude and outdated bank switching arrangements is dramatic.

The central processor runs in both user and supervisor states. It can detect and reject a defective user program. It is impossible for a user program to go bad and stop the entire system, as can happen quite easily in less sophisticated systems.

Task switching is accomplished by use of a multiple map RAM memory, with sixty-four individual task maps. Each task can access from 4 to 64 K-bytes of memory. Multiple tasks may be used in programs that require more than 64K bytes of memory for execution. When a task is completed the memory is automatically released for other use.

SOFTWARE

The S+ operating system, UniFLEX® is a multiuser, multitasking operating system based on the UNIX® operating system that has been used for many years on Digital Equipment Corp. PDP-11 series minicomputers. It is considered one of the most sophisticated and "user friendly" operating systems available. Variations of UNIX® are rapidly becoming standard on mini and larger microcomputers.

A large variety of languages are available for use with the system. These include FORTRAN, COBOL, BASIC, and Pascal. Word processing packages are also available to give you full text processing capability on the system.

Applications programs are available in large quantities in many fields. This includes general business, medical, dental, veterinary, library and real estate management; plus others. Since the system is multiuser it can also be connected to cash registers to produce a point-ofsale terminal system combined with the computer. The possibilities for application of this system are endless.

THE I/O SYSTEM

The S+ system is totally interrupt driven. All terminal and printer I/O devices connect to an I/O bus separate from the main bus. Up to thirty-two separate devices may be connected to the I/O bus at any one time. If I/O activity is great enough to cause an unacceptable slowdown in system operation, a separate I/O processor can be installed in the system. This plug-in option removes all I/O handling overhead from the main processor and allows operation of up to thirty-two external devices at 9,600 baud. Without an integrated total design, as in the S+ system, it would become impractical to use a UNIX®type operating system in a situation with heavy terminal I/O activity.

DISK STORAGE

A wide range of disk storage capacity is available for the S+ system, from 2.5 M-byte floppy disks to an 80 M-byte Winchester and many sizes between. All disk controllers use direct memory access (DMA) type operations to maximize data transfer and to minimize overhead on the main processor. The Winchester disks also use intelligent controllers along with DMA transfers to preserve the performance that these type devices are capable of giving. Without this distributed intelligence the system performance would be greatly degraded. The UniFLEX®operating system is designed to work at maximum efficiency with this type disk system. The data transfer rates achieved by this combination rival those of large minicomputers.

COMMUNICATIONS

A high speed local network communications system is available to interconnect S+ systems. The VIABUS® network will allow communication between systems at data rates of over 400K baud. Such a system makes it possible to share data between local systems in an efficient and low-cost manner.

AVAILABLE SOON

Tape backup—20M-Byte in less than 15 minutes on a standard ¼ inch cartridge.

Mini-Wini—5 and 10 M-Byte Winchesters—5¼ inch package. Winchester performance, for smaller systems in a small package. UniFLEX® compatible design.

Large Capacity—190 and 340 M-Byte Winchesters, plus SMD cartridge drives.

UniFLEX is a registered trademark of Technical Systems Consultants, Inc.

UNIX is a registered trademark of Bell Labs.

VIABUS is a registered trademark of Southwest Technical Products Corporation.

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 **(512) 344-0241**

```
                        * FATAL DOUBLE ERROR
C2E3 8E  C52F           FATAL   LDX   #FATAL
C2E6 BD  C505                   JSR   FERR

C2E9 F8  C10F   FC320   STB   TRKSEC    SET BACKUP
C2EC 33  5F             LEAU  -1,U     CORRECT SECTOR COUNT
C2EE 7E  C265           JMP   FC205     REWRITE LAST SECTOR
C2F1 FC  C10D   FC340   LDD   END_TS
C2F4 F8  C10F           STD   TRKSEC    UPDATE CURRENT
C2F7 7E  C25F           JMP   FC200     GET NEW AND REWRITE


                        * FINDS NEXT FREE SECTOR IN MAP POINTED BY Y
                        * ENTERED WITH D=CURRENT TRK/SEC EXIT D=NEXT FREE TRK/SEC
                        * B=0 IF AT END, X DESTORYED, U PRESERVED
C2FA 5C                 FNDFRE  INCB            SECTOR
C2FB F1  C104           CMPB    MAITRK+1        CK FOR NEW TRACK
C2FE 23  08             BLS     FND100          NO
C300 C6  01             LDB     #1              RESET SECTOR 0
C302 4C                 INCA                    NEXT TRACK
C303 B1  C103           CMPA    MAITRK
C306 22  05             BHI     FND180          PAST LAST TRACK

C308 AE  A1     FND100  LDX     ,Y++            CK FOR FREE
C30A 26  EE             BNE     FNDFRE          NO. LOOP
C30C 39                 RTS

C30D CC  0000   FND180  LDD     #0              END
C310 39         FND160  RTS

                        * READS ONE SECTOR
                        * X=> FCB  D=TRACK/SECTOR
C311 ED  88 1E  READ1   STD     30,X            SET TRACK/SECTOR
C314 86  09             LDA     #9
C316 A7  84             STA     ,X              SINGLE SECTOR READ
C318 7E  B408           JMP     FMSCAL          READ AND RETURN

                        * MARKS FILES IN MAPS, CHECKS FOR ERROR IN
                        * RECORD NUMBER, FILES LINKED TOGETHER AND
                        * VERIFIES ENDING SECTOR
C31B CC  0000   TRACE   LDD     #0
C31E FD  C100           STD     PRV_TS          FLAG LAST SECTOR
C321 FD  C105           STD     RECORD          RESET RECORD COUNT

C324 7D  C10A   TRCLOP  TST     SECPRT          CK FOR PRINT ALL SEC
C327 27  06             BEQ     TRC10           NO
C329 8E  C10F           LDX     #TRKSEC
C32C BD  C4F4           JSR     PRTSEC
C32F 8E  CB40   TRC10   LDX     #MRKFCB
C332 FC  C10F           LDD     TRKSEC          GET CURRENT TRACK/SECTOR
C335 5D                 TSTB                    CK FOR ZERO SEC
C336 27  0A             BEQ     TRC20           INVALID
C338 B1  C103           CMPA    MAITRK          TRACK IN RANGE
C33B 22  05             BHI     TRC20           NO
C33D F1  C104           CMPB    MAITRK+1        CK SECTOR
C340 23  06             BLS     TRC40           OK

C342 8E  C5EB   TRC20   LDX     #BADSEC         GIVE BAD TRACK/SECTOR
C345 7E  C407           JMP     TRCERR          REPORT ERROR AND EXIT

C348 BD  C7     TRC40   BSR     READ1           GET NEXT SECTOR
C34A 27  09             BEQ     TRC60           READ OK

C34C B0  CB3F           JSR     RPTERR          GIVE ERROR TYPE
C34F 8E  C5FB           LDX     #RDSEC
C352 7E  C407           JMP     TRCERR

                        * CK FOR RECORD COUNT ERR
C355 7D  C111   TRC60   TST     DIRMAP          CK FOR DIRECTORY
C358 2B  41             BMI     TRC100          YES, NO RECORD NUMBERS
C35A 7D  C107           TST     RFLAG           CK FOR RANDOM FILE
C35D 27  2D             BEQ     TRC80           NO

                        * IST 2RECORDS AREECIAL IN RANDOM FILES
C35F 7A  C107           DEC     RFLAG           MARK COUNT
C362 FC  C113           LDD     SECNT           DEC COUNT FOR RANDOM RECORD
C365 83  0001           SUBD    #1
C368 FD  C113           STD     SECNT
C36B EC  B8 42          LDD     $40+2,X         GET RECORD COUNT
C36E 27  2B             BEQ     TRC100          RANDOM RECORD SHOULD BE ZERO

                        * REPORT RECORD # ERROR
C370 8E  C610   RECERR  LDX     #RECN
C373 BD  CD1E           JSR     PSTRNG
C376 8E  CB02           LDX     #MRKFCB+$40+2 POINT TO RECORD NUMBER
C379 BD  CD45           JSR     OUTADR
C37C B0  C4C9           JSR     PSPC
C37F 8E  C105           LDX     #RECORD         POINT TO CORRECT NUMBER
C382 BD  CD45           JSR     OUTADR
C385 8E  C605           LDX     #SECN
C388 BD  7D             BSR     TRCERR          FINISH ERROR MESSAGE
C38A 20  0F             BRA     TRC100

C38C FC  C105   TRC80   LDD     RECORD          UPDATE CURRENT NUMBER
C38F C3  0001           ADDD    #1
C392 FD  C105           STD     RECORD
C395 10A3 B8 42         CMPD    $40+2,X         COMPARE WITH # IN FILE
C399 26  05             BNE     RECERR          NO GIVE ERROR

                        * PUT ENTRY IN MAP
                        * COMPUTE LOCATION IN MAP
C39B B6  C10F   TRC100  LDA     TRKSEC          GET TRACK #
C39E F6  C104           LDB     MAITRK+1        GET MAI # OF SECTORS
C3A1 3D                 MUL
C3A2 FB  C110           ADDB    TRKSEC+1        OFSET FOR THIS SECTOR
C3A5 89  00             ADCA    #0
C3A7 58                 ASLB                    *2
C3A8 49                 ROLA
C3A9 25  36             BCS     TRC220          OVERFLOW
C3AB C3  027E           ADDD    #MAP-2
C3AE 25  31             BCS     TRC220
C3B0 1083 CC2B          CMPD    #MAPEND         ENOUGH ROOM
C3B4 24  2B             BHS     TRC220          NO

C3B6 1F  02             TFR     D,Y             POIN TO MAP
C3B8 EC  A4             LDD     ,Y              CK FOR CURRENT ENTRY
C3BA 27  20             BEQ     TRC200          OK

C3BC 8E  C628           LDX     #FILC
C3BF BD  CD1E           JSR     PSTRNG          PRINT COLLISION MSG
C3C2 6D  A4             TST     ,Y              CK FOR DIRECTORY FILE
C3C4 2A  0A             BPL     TRC150          NO
C3C6 8E  C63B           LDX     #DIR            PRINT DIRECTORY
C3C9 C6  09             LDB     #9              MSG LENGHT
C3CB B0  C4B6           JSR     PMAR
C3CE 20  07             BRA     TRC160          CONT ERROR MSG
C3D0 1F  21     TRC150  TFR     Y,X             POINT TO DTR #
C3D2 C6  04             LDB     #4
C3D4 BD  C4EA           JSR     OUTSEC          PRINT COLLISION #
C3D7 8E  C605   TRC160  LDX     #SECN
C3DA 20  2B             BRA     TRCERR          FINISH ERROR MSG

                        * CK FOR NEXT LINK
C3DC FC  C111   TRC200  LDD     DIRMAP          SET IN MAP
C3DF ED  A4             STD     ,Y
C3E1 8E  CB00   TRC220  LDX     #MRKFCB+$40
C3E4 10BE C10F          LDY     TRKSEC          GET CURRENT T/S
C3E8 EC  84             LDD     ,X
C3EA 27  0A             BEQ     TRC300          LAST SECTOR
C3EC 10BF C10D          STY     PRV_TS          UPDATE PREVIOUS
C3F0 FD  C10F           STD     TRKSEC          NEW CURRENT TRK/SEC
C3F3 7E  C324           JMP     TRCLOP          DO AGAIN

                        * CK FOR ENDING NOT MATCHING WITH DIRECTORY
C3F6 7D  C111   TRC300  TST     DIRMAP
C3F9 2B  08             BMI     TRC400          BYPASS FOR DIR
C3FB 10BC C10B          CMPY    END_TS
C3FF 27  05             BEQ     TRC400

C401 8E  C648           LDX     #ENDERR         GIVE END ERROR
C404 BD  01             BSR     TRCERR

C406 39         TRC400  RTS

                        * PRINT TRACE ERROR MESSAGE
C407 BD  CD1E   TRCERR  JSR     PSTRNG
C40A 8E  C10F           LDX     #TRKSEC         GIVE FAILING TRACK/SEC
C40D BD  CD45           JSR     OUTADR
C410 8E  C659           LDX     #PRVSEC         PRINT PREVIOUS TRACK/SEC
C413 BD  CD1E           JSR     PSTRNG
C416 8E  C10D           LDX     #PRV_TS
C419 B0  CD45           JSR     OUTADR
C41C 39                 RTS
```

```
                        * PRINT S.I.A.
C41D 8E  C69C   PSIA    LDX     #SIAHDR
C420 BD  CD1E           JSR     PSTRNG
C423 CC  FFFF           LDD     #$FFFF          FLAG SIR MODE
C426 FD  C111           STD     DIRNUM          AND SET FOR DIR TRACE
C429 8E  0190           LDX     #SIAFCB+$40+16 POINT TO SIR DATA

                        * PRINT DISK NAME OR DIRECTORY POINTED BY X
C42C BD  CD24   PRTNAM  JSR     PCRLF
C42F 7D  C111           TST     DIRNUM          CK FOR SIR
C432 2B  0F             BMI     PNLO            YES
C434 34  10             PSHS    X
C436 8E  C111           LDX     #DIRNUM
C439 C6  04             LDB     #4
C43B BD  C4EA           JSR     OUTDEC          PRINT DIRECTORY #
C43E BD  C4C7           JSR     PSPC2
C441 35  10             PULS    X

C443 C6  08     PNLO    LDB     #8
C445 34  10             PSHS    X               SAVE POINTER
C447 8D  60             BSR     PNAM            PRINT NAME
C449 7D  C111           TST     DIRNUM          DIR?
C44C 2B  05             BMI     PN20
C44E 86  2E             LDA     #'.
C450 BD  CD1B           JSR     PUTCHR
C453 C6  03     PN20    LDB     #3
C455 8D  5F             BSR     PNAM            PRINT EXTENSION

C457 7D  C111           TST     DIRNUM
C45A 2A  09             BPL     PN30
C45C C6  05             LDB     #5
C45E BD  C4E8           JSR     DEC5            PRINT VOL#
C461 8D  62             BSR     PSPC4
C463 20  09             BRA     PN40
C465 8D  60     PN30    BSR     PSPC2
C467 8D  CD3C           JSR     OUTHEX          PRINT PROTECTION
C46A 8D  58             BSR     PSPC2
C46C 30  02             LEAX    2,X

C46E BD  C4F4   PN40    JSR     PRTSEC          PRINT STARTING TRACK/SEC
C471 BD  C4F4           JSR     PRTSEC          ENDING T/S
C474 C6  04             LDB     #4
C476 8D  C4EA           JSR     OUTDEC          PRINT SIZE
C479 7D  C111           TST     DIRNUM
C47C 2B  0D             BMI     PN50
C47E 8D  47             BSR     PSPC2
C480 86  53             LDA     #'S             ASSUME SEQUENTIAL FILE
C482 6D  81             TST     0,X++
C484 27  02             BEQ     PN45            YES
C486 86  52             LDA     #'R             NO RANDOM
C488 8D  CD1B   PN45    JSR     PUTCHR

C48B 8D  3C     PN50    BSR     PSPC
                        * PRINT DATE
C48D 8D  3F             BSR     PNUM            MONTH
C48F 86  2D             LDA     #'-
C491 8D  CD1B           JSR     PUTCHR
C494 8D  38             BSR     PNUM            DAY
C496 86  2D             LDA     #'-
C498 8D  CD1B           JSR     PUTCHR
C49B 8D  31             BSR     PNUM            YEAR

C49D 7D  C111           TST     DIRNUM
C4A0 2A  04             BPL     PN55            NOT SIR
C4A2 8D  23             BSR     PSPC2
C4A4 8D  4E             BSR     PRTSEC          GIVE NXT T/S

C4A6 35  10     PN55    PULS    X               CK FOR DELETE
C4A8 6D  84             TST     ,X
C4AA 2A  07             BPL     PN60            NO
C4AC 8E  C6D7           LDX     #DEL
C4AF C6  07             LDB     #7
C4B1 8D  03             BSR     PNAM
C4B3 7E  CD24   PN60    JMP     PCRLF

                        * PRINT STRING POINTED BY X FOR B CHAR
C4B6 A6  80     PNAM    LDA     ,X+
C4B8 2B  02             BMI     PNAM5           DELETED NAME
C4BA 26  02             BNE     PNAM10          NULL
C4BC 86  20     PNAM5   LDA     #$20
C4BE BD  CD1B   PNAM10  JSR     PUTCHR
C4C1 5A                 DECB
C4C2 26  F2             BNE     PNAM
C4C4 39                 RTS

C4C5 8D  00     PSPC4   BSR     PSPC2
C4C7 8D  00     PSPC2   BSR     PSPC
C4C9 86  20     PSPC    LDA     #$20
C4CB 7E  CD1B           JMP     PUTCHR

                        *PRINT 2 CHAR DEC NUMBER POINTED BY X
C4CE E6  80     PNUM    LDB     ,X+             SET NUM
C4D0 4F                 CLRA                    COUNTER

C4D1 C0  0A     PNUM10  SUBB    #10
C4D3 25  03             BCS     PNUM20
C4D5 4C                 INCA
C4D6 20  F9             BRA     PNUM10

C4D8 CB  0A     PNUM20  ADDB    #10             RESTORE
C4DA 34  04             PSHS    B               SAVE REMAINDER
C4DC 8B  30             ADDA    #'0             MAKE NUMBER
C4DE BD  CD1B           JSR     PUTCHR
C4E1 35  02             PULS    A               GET REMAINDER
C4E3 8B  30             ADDA    #'0
C4E5 7E  CD1B           JMP     PUTCHR


                        * FIX FOR FLEX OUTDEC
C4E8 C6  05     DEC5    LDB     #5              PRINT 5 CHAR
C4EA 34  10     OUTDEC  PSHS    X
C4EC BD  CD39           JSR     FLXDEC
C4EF 35  10             PULS    X
C4F1 30  02             LEAX    2,X
C4F3 39                 RTS

                        * PRINT TRACK/SECTOR POINTED BY X
C4F4 BD  CD3C   PRTSEC  JSR     OUTHEX          PRINT TRACK
C4F7 30  01             LEAX    1,X
C4F9 86  2F             LDA     #'/
C4FB BD  CD1B           JSR     PUTCHR
C4FE BD  CD3C           JSR     OUTHEX          SECTOR
C501 30  01             LEAX    1,X
C503 20  C4             BRA     PSPC

C505 BD  CD1E   ERR     JSR     PSTRNG          PRINT ERROR MESSAGE
C508 7E  C003           JMP     WARMS           AND EXIT

C508 86  FF     PROMPT  LDAA    #$FF            POINT TO CRT
C50D B7  CC22           STA     SWITCH
C510 BD  CD1E           JSR     PSTRNG          PRINT REQUEST
C513 BD  CD15           JSR     GETCHR          GET RESPONSE
C516 84  5F             ANDA    #$5F            FORCE UPPERCASE
C518 7F  CC22           CLR     SWITCH          POINT TO PRINTER
C51B 81  59             CMPA    #'Y             CK FOR YES
C51D 39                 RTS

C51E 42 41 44 20 IORV   FCC     "BAD DRV #",4
C528 42 41 44 20 BSIR   FCC     "BAD SIR"
C52F 2D 20 46 41 FATAL  FCC     "- FATAL !",4
C539 4D 41 58 20 TS_MSG FCC     "MAX TRK/SEC "
C545 4F 4B 20 59 OK     FCC     "OK(Y/N)",4
C54D 4C 49 54 27 MEMWRN FCC     "LIT'D MEMORY-ALL SECTORS"
C565 20 4E 4F 54        FCC     " NOT CK'D",$D,$A
C570 2A 2A 20 44        FCC     "** DON'T CONSOLIDATE FREE CHAIN **",4
C593 50 52 49 4E SECMDE FCC     "PRINT EACH LINK Y/N?",4
C5A8 46 52 45 45 FCNUM  FCC     "FREE SECTORS-IN MAP-IN SIR",4
C5C3 43 4F 4E 53 FCPRMT FCC     "CONSOLIDATE FREE CHAIN Y/N?",4
C5DF 46 41 54 41 FDIR   FCC     "FATAL DIR ERR",4
C5ED 49 4E 56 41 BADSEC FCC     "INVALID TRK/SEC",4
C5FD 52 44 20 45 RDSEC  FCC     "RD ERR-",4
C605 41 54 20 53 SECN   FCC     "AT SECTOR-",4
C610 52 45 43 4F RECN   FCC     "RECORD #ERR "
C61C 57 41 53 2F        FCC     "WAS/SHOULD BE ",4
C628 43 4F 4C 4C FILC   FCC     "COLLISION WITH # ",4
C638 44 49 52 45 DIR    FCC     "DIRECTORY-",4
C648 45 4E 44 20 ENDERR FCC     "END MISMATCH AT-",4
C659 50 52 56 53 PRVSEC FCC     "PRVS TRK SEC-",4
C667 46 49 4C 45 DIRHDR FCC     "FILE#      NAME      PRT  $E$"
C683 20 45 4E 44        FCC     " END     SIZE TYPE  DATE",4
C69C 44 49 53 4B SIAHDR FCC     "DISK NAME   VOL"
C6AB 20 20 20 20        FCC     "    F.C.START END  SIZE "
```

```
C6C3 20 44 41 54          FCC   ' DATE    MAI TRKSEC',4
E6D7 2A 44 4C 54   DEL    FCC   '*DLT'01'

                          END   START
```

BIT BUCKET

DEAR DON,

HERE IS A HANDY PROGRAM QUICKY FOR 6800 XBASIC:

XBERPACH.ASM 5-30-82 PAGE 1

```
                * PATCH TO TSC 6800 EXTENDED BASIC (X-BASIC)
                * TO CAUSE ERROR FILE TO BE READ WHEN ERROR OCCURS.
                * MODIFY ERRORS.SYS TO INCLUDE THE ERRORS FROM
                * XBASIC. (SEE MAY 82 ISSUE OF '68' MICRO, PAGE 33)
                * THIS PATCH DOES THE SAME FOR 6800 XBASIC AS DOES
                * DR. LAWRENCE PRIBLE'S PATCH FOR 6809 XBASIC.
                *
                * EXTERNAL REFERENCES:
AD24            PCRLF   EQU   $AD24
AD3F            RP ERR  EQU   $AD3F
                *
                * PATCH IS PLACED IN MEMORY WHERE XBASIC WAS ABOUT
                * TO PRINT "ERROR #" -- (NOT NEEDED NOW!)
0C4C                    ORG   $0C4C
0C4C BD AD 24           JSR   PCRLF
0C4F CE 01 22           LDX   #$0122      ERROR CODE STORAGE-1
0C52 5F                 CLR B
0C53 F7 01 26           STA B $0126
0C56 BD AD 3F           JSR   RPTERR      LOCATE & PRINT ERROR STRING
0C59 BD AD 24           JSR   PCRLF

                        END

NO ERROR(S) DETECTED

THE ORIGINAL XBASIC CODE WAS:

0C4C CE 0C98
0C4F BD 0401
0C52 CE 0122
0C55 5F
0C56 F7 0126
0C59 BD 106E
```

SUBMITTED BY

ROGER S. GUSTERHOUT
511 SEYMOUR ST.
OGDENSBURG, NY 13669

THE HAHNEMANN MEDICAL COLLEGE & HOSPITAL OF PHILADELPHIA
TWO-THIRTY NORTH BROAD STREET PHILADELPHIA, PENNSYLVANIA 19102

DIVISION OF PULMONARY DISEASES

May 19, 1982

Don Williams
5900 Cassandra Smith
P.O. Box 849
Hixson, TN 37343

Dear Don,

I have several software incompatibility problems that per aps have been encountered by other readers:

1. When my system is called up remotely through my Thomas modem, several essential programs such as the SWTP editor and RMS will not run, as control is transferred to the local terminal rather than through the modem. My system is a SWTP 56K 6809.

2. When processing texts (SWTP FLEX 2.01, TSC PR 4, and SWTP editor 3.1:2), one or two words will be dumped at the bottom of a page on the left hand side after the last allowable formatted line. I wonder if anyone knows how to get around this problem.

3. When text is entered in the TSC editor (line oriented) and later worked on with the SWTP editor (the one that highlights), long lines are truncated. Is there a simpler way of running a text to limit line length other than manually chanding line by line. If we forget to RETURN at the end of the screen?

4. Finally, the SWTP editor does not allow you to manipulate the special characters used in the TSC PR.

Sincerely,

Allen P. Freedman, M.D.
Associate Professor of Medicine

APF/jco

Computer Systems Consultants, Inc.
1454 Latta Lane, Conyers, GA 30207
Telephone Number 404-483-1717/4570

We are pleased to announce the availability of two new products and the ability to run all CSC products under both FLEX and UNIFLEX.

Our new products are as follows:

6800-6809 AND 6809 PIC/PID XLATORS both $50-FLEX $60-UNIFLEX
These programs enable the user to translate 6800/1 assembler-language programs to use 6809 mnemonics and to convert 6809 assembler-language programs to position-independent code and data; they are especially useful in the conversion to OS/9 Level 1.

UNIFLEX SIMULATOR FOR FLEX $100-FLEX $110-UNIFLEX
This program enables the user to debug UNIFLEX assembler language programs using the TSC DEBUG and other facilities of FLEX.

Our other products are as follows:

SUPER SLEUTH DISASSEMBLER $99-FLEX $100-UNIFLEX

Z-80/8080/5 SUPER SLEUTH DISASSEMBLER $99-FLEX $100-UNIFLEX

CROSS-ASSEMBLERS each $50,3/$100-FLEX each $60,5/$120-UNIFLEX

6805 and 6502 DEBUGGING SIMULATORS each $75-FLEX $80-UNIFLEX

6502-TO-6809 XLATOR SYSTEM $75-FLEX $80-UNIFLEX

FULL SCREEN FORMS DISPLAY (6809 X-BASIC) $50-FLEX $75-UNIFLEX

FULL SCREEN MAILING LIST (6809 X-BASIC) $100-FLEX $110-UNIFLEX

FULL SCREEN INVENTORY/MRP (6809 X-BASIC) $100-FLEX $150-UNIFLEX

TABULA RASA SPREADSHEET (6809 X-BASIC) $100-FLEX $200-UNIFLEX

TSC BASIC/XPC UTILITY PROGRAMS all $25-FLEX $50-UNIFLEX
Please note that all BASIC utilities (RESEQ, XREF, and DISK SORT) are now all included in one package, with one price for all three.

All our programs are provided in source format on disk.
Detailed printed manuals are provided with all products.
Please call or write for catalog and dealer information.

FLEX and UNIFLEX are trademarks of Technical Systems Consultants.

.... 6809 ENTHUSIASTS AND OS-9 USERS ....

A users' group for those interested in the 6809 processor, the OS-9 operating system, and UNIX-like systems on 68XX machines is forming. You are invited to participate with your ideas and other contributions.

The formation of the group was initiated at a meeting in Des Moines on May 16. Present activities are centered around the establishment of a set of broad-based goals for the group. Provisional officers were elected in Des Moines and are as follows:

President	Brian Capouch	RR #1 Box 270 Hoosa [illegible] 47949
Vice-President	Shel Epstein	Box 409 Wilmette, IL 60091
Secretary	Howard Markosis	P.O. Box 28956 Dallas, TX 75228
Treasurer	Erwin Strachley	1003 Roble Lane Santa Barbara, CA 93103

A Bulletin Board System has been temporarily established to foster the organization of a users' group and to share information between users of the OS-9 operating system. If you have an ASCII terminal and a 300 baud modem, you may use this public information utility by following these instructions:

Dial (312) 397-8308 This is in the Chicago suburbs.

When a connection is established, alternately press the Carriage Return key and the Line Feed keys (CTRL-M and CTRL-J) on the keyboard. Repeat until you receive the message: PLEASE LOG IN

Respond to this request with: HELLO-G500,,) This must be typed exactly as written here. No extra

spaces or other changes may be made. You should then be able to follow the instructions given to read and send messages.

Please leave the system formally, not just by hanging up. You will have an opportunity to leave suggestions or other information to the operators of the system.

Initially, the goals of the group will be to provide the following services:

1. A Chat/Mail utility for users
2. A users' interest survey and alias
3. A commercial software registry
4. A public domain software library
5. A periodical in print or electronic media for information sharing

Those interested in participating in the formation of a users' group for OS-9 users should indicate their interest to one of the provisional officers or via the bulletin board.

The OS-9 operating system for the 6809 processor is a product of Microware Systems Corporation. The users' group presently forming is entirely independent of the Microware company.

COMPUSENSE LTD.
Computer Systems Consultants

P.O Box 169
London N13 4HT
Tel: 01-882 0681

Mr. Dan Williams
68 MicroJournal
5900 Cassandra Smith
P.O. Box 849
Hixson , TN 37343

18th June 1982

Dear Dan ,

As it is Ascot week her in England , I thought you might like to publish the enclosed listing of a mini Ascot race for your readers . The program is written in TSC XBASIC for a SWTPC CT-82 terminal , and therefore some control codes would need to be altered for other terminals.

Hope you pick a winner !

Yours sincerely ,

Ted Opyrchal.

```
1 REM !!!!!!! ASCOT HORSE RACE !!!!!!!
2 REM NG = NUMBER OF HAGS
3 REM PS(NG) = CURRENT POSITION OF EACH HORSE
4 REM W   = WINNER COUNT
5 REM CL$ = HOME & CLEAR SCREEN
6 REM C1$ = CURSOR ON
7 REM C2$ = CURSOR OFF
8 REM CS$ = CURSOR SET CODE (Y,X)
9 REM IN$ = INSERT CHARACTER IN LINE CODE
10 REM
11 REM  THIS GAME CONFIGURED FOR SWTPC CT-82 TERMINAL
12 REM
13 REM   COMPUSENSE LTD. , P.O. BOX 169 LONDON N13 4HT
14 REM
15 ON ERROR GOTO 30
20 GOTO 40
30 RESUME 90
40 DATA "\___/|"
50 DATA "/   \"
60 DATA "----------
70 NG=6
80 DIM PS(NG),H$(3)
90 CL$ = CHR$(16)+CHR$(22)
100 C1$ = CHR$(21) : REM CURSOR ON
110 C2$ = CHR$(5)  : REM CURSOR OFF
120 PRINT C2$;
130 PRINT CL$;
140 RESTORE 40
150 FOR J = 1 TO 3
160 READ H$(J)
170 NEXT J
180 IN$ = CHR$(28)+CHR$(24)
190 CS$ = CHR$(28)+CHR$(11)
200 PRINT CHR$(36);
210 PRINT "TYPE 0 TO QUIT OR 9 TO START A RACE ";
220 Y$ = INCH$(0)
230 IF Y$ = "0" THEN PRINT CL$;C1$;:END
240 IF Y$ <> "9" THEN 200
250 PRINT CL$;
260 W = 0
270 FOR X = 1 TO NG
280 PS(X) = 7
290 PRINT H$(1);:PRINT H$(2)
300 IF X < NG THEN PRINT H$(3)
310 NEXT X
320 I = INT(RND(0)*NG+1)
330 IF PS(I) < 7 THEN 320
340 L1 = INT(RND(0)*2+1)
350 FOR L = 1 TO L1
360 FOR J = 0 TO 1
370 Y = J+(I-1)*3
380 C0=CS$+CHR$(Y)+CHR$(0)
390 PRINT C0;IN$;" ";
400 NEXT J
410 NEXT L
420 PS(I)=PS(I)+L
430 IF PS(I) < 82 THEN 320
440 PS(I) = 0
450 W = W+1
460 Y = (I-1)*3+1
470 C0 = CS$+CHR$(Y)+CHR$(36)
480 PRINT C0;
490 PRINT "HORSE ";I;
500 IF W = 1 THEN PRINT "THE WINNER" ;
510 IF W = 2 THEN PRINT "PLACE" ;
520 IF W = 3 THEN PRINT "SHOW" ;
530 IF W < 3 THEN 320
540 W = 0
550 FOR J = 1 TO NG : PS(J) = 0 : NEXT J
560 GOTO 140
```

MRL ectronics b.v.
The very first computershop in Holland.

Telefoon (015) 56 92 88*
Postbus 98, 2800 AB Delft

Telex 38344

68 Micro Journal,
Mr. Don Williams,
P.O. Box 849,
Hixson, TN 37343,
United States.

Ons kenmerk: 103-2-jvd Delft, April 12, 1982

Dear Don,

Being the dealer of SWTPc and TSC in this part of Europe from 1975 we found that the problem with most of the data base managers for UniFLEX* systems is that they are just FLEX* programs converted to UniFLEX.

That's why we decided to design a new Data Base Manager for UniFLEX systems. Now we could use (and we did) the neat trick of UniFLEX and that made life much easier, i.e. foreground and background tasks, just to mention one of the most important.
We declared it as a goal of design to make it possible to have the availability to use the same data base with several users at the same time.

Now it is possible to define your own records without trying to get all the info in just 252 bytes, to define your own reports with the selections you want ("and" and "or" and nested as well) and to keep them on disk. You want to print records? Just give the report number, the number of copies and the output device of all the reports you want. After that, the system will start processing and you can go on with an other job at the same time. That's real multi-user!

We did not design a sort/merge as we could not do better and faster than the TSC sort/merge can do. One needs to buy that with TSC.

Thanks, Don, for the opportunity,

Sincerely yours,

Jaap van Duffelen.

---

* UniFLEX is a TSC trademark

William Hartmann
RR 2 Box 121-1
Blue Springs, MO 64015

April 12, 1982

COMMENTS ON FLEX DISK FORMAT By LEO TAYLOR
April 1982, 68 Micro Journal

In bringing up my system using the general version of flex I found the same things about the speed of FLEX as reported in by Mr. Taylor. One way to speed up the write operation is to use a bit map of the next available sector instead of reading the sector first. The bit map will have a bit set for every free sector.

```
                        * FATAL DOUBLE ERROR
C2E3 8E C32F            LDX   #FATAL
C2E6 BD C505            JSR   FERR

C2E9 FD C10F    FC320   STD   TRKSEC    SET BACKUP
C2EC 33 9F              LEAU  -1,U      CORRECT SECTOR COUNT
C2EE 7E C265            JMP   FC205     REWRITE LAST SECTOR
C2F1 FC C108    FC340   LDD   END_TS
C2F4 FB C10F            STD   TRKSEC    UPDATE CURRENT
C2F7 7E C25F            JMP   FC200     GET NEW AND REWRITE


                        * FINDS NEXT FREE SECTOR IN MAP POINTED BY Y
                        * ENTERED WITH B=CURRENT TRK/SEC EXIT B=NEXT FREE TRK/SEC
                        * D=0 IF AT END, X DESTORYED, U PRESERVED
C2FA 5C                 FNBFRE INCB            SECTOR
C2FB F1 C104            CMPB  MAXTRK+1  CK FOR NEW TRACK
C2FE 23 08              BLS   FNB100    NO
C300 C6 01              LDB   #1        RESET SECTOR 0
C302 4C                 INCA            NEXT TRACK
C303 B1 C103            CMPA  MAXTRK
C306 22 05              BHI   FNB180    PAST LAST TRACK

C308 AE A1      FNB100  LDX   ,Y++      CK FOR FREE
C30A 26 EE              BNE   FNBFRE    NO, LOOP
C30C 39                 RTS

C30D CC 0000    FNB180  LDD   #0        END
C310 39         FNB160  RTS

                        * READS ONE SECTOR
                        * I=> FCB  B=TRACK/SECTOR
C311 ED 88 1E   READI   STD   30,X      SET TRACK/SECTOR
C314 86 09              LDA   #9
C316 A7 84              STA   ,X        SINGLE SECTOR READ
C318 7E 0406            JMP   FRSCAL    READ AND RETURN

                        * MARKS FILES IN MAPS, CHECKS FOR ERROR IN
                        * RECORD NUMBER, FILES LINKED TOGETHER AND
                        * VERIFIES ENDING SECTOR
C31B CC 0000    TRACE   LDD   #0
C31E FD C108            STD   PRV_TS    FLAG LAST SECTOR
C321 FD C105            STD   RECORD    RESET RECORD COUNT

C324 7D C10A    TRCLOP  TST   SECPNT    CK FOR PRINT ALL SEC
C327 27 06              BEQ   TRC10     NO
C329 8E C10F            LDX   #TRKSEC
C32C BD C4F4            JSR   PRTSEC
C32F 8E C800    TRC10   LDX   #WRKFCB
C332 FC C10F            LDD   TRKSEC    GET CURRENT TRACK/SECTOR
C335 5D                 TSTB            CK FOR ZERO SEC
C336 27 04              BEQ   TRC20     INVALID
C338 B1 C103            CMPA  MAXTRK    TRACK IN RANGE
C33B 22 05              BHI   TRC20     NO
C33D F1 C104            CMPB  MAXTRK+1  CK SECTOR
C340 23 06              BLS   TRC40     OK

C342 8E C3E0    TRC20   LDX   #BADSEC   GIVE BAD TRACK/SECTOR
C345 7E C407            JMP   TRCERR    REPORT ERROR AND EXIT

C348 BD C7      TRC40   BSR   READI     GET NEXT SECTOR
C34A 27 09              BEQ   TRC60     READ OK

C34C BD CD3F            JSR   RPTERR    GIVE ERROR TYPE
C34F 8E C3FD            LDX   #WRNSEC
C352 7E C407            JMP   TRCERR

                        * CK FOR RECORD COUNT ERR
C355 7D C111    TRC60   TST   DIRFLG    CK FOR DIRECTORY
C358 2B 48              BMI   TRC100    YES, NO RECORD NUMBERS
C35A 7D C107            TST   RFLAG     CK FOR RANDOM FILE
C35D 27 20              BEQ   TRC80     NO

                        * IST 2RECORDS AREECIAL IN RANDOM FILES
C35F 7A C107            DEC   RFLAG     MARK COUNT
C362 FC C113            LDD   SECNT     DEC COUNT FOR RANDOM RECORD
C365 83 0001            SUBD  #1
C368 FD C113            STD   SECNT
C36B EC 88 42           LDD   #40+2,X   GET RECORD COUNT
C36E 27 2B              BEQ   TRC100    RANDOM RECORD SHOULD BE ZERO

                        * REPORT RECORD 0 ERROR
C370 8E C610    RECERR  LDX   #RECK
C373 BD CD1E            JSR   PSTRNG                    *
C376 8E C882            LDX   #WRKFCB+#40+2 POINT TO RECORD NUMBER
C379 BD CD45            JSR   OUTADR
C37C BD C4C9            JSR   PSPC
C37F 8E C105            LDX   #RECORD   POINT TO CURREC NUMBER
C382 BD CD45            JSR   OUTADR
C385 8E C605            LDX   #SECK
C388 8D 7D              BSR   TRCERR    FINISH ERROR MESSAGE
C38A 20 0F              BRA   TRC100

C38C FC C105    TRC80   LDD   RECORD    UPDATE CURRENT NUMBER
C38F C3 0001            ADDD  #1
C392 F0 C105            STB   RECORD
C395 10A3 88 42         CMPD  #40+2,X   COMPARE WITH 0 IN FILE
C399 26 05              BNE   RECERR    NO GIVE ERROR

                        * PUT ENTRY IN MAP
                        * COMPUTE LOCATION IN MAP
C398 86 C10F    TRC100  LDA   TRKSEC    GET TRACK #
C39E F6 C104            LDB   MAXTRK+1  GET MAX # OF SECTORS
C3A1 30                 MUL
C3A2 FB C110            ADDB  TRKSEC+1  OFSET FOR THIS SECTOR
C3A5 89 00              ADCA  #0
C3A7 58                 ASLB            #2
C3A8 49                 ROLA
C3A9 25 36              BCS   TRC220    OVERFLOW
C3AB C3 027E            ADDD  #MAP-2
C3AE 25 31              BCS   TRC220
C3B0 10B3 CC2B          CMPD  #ENDMAP   ENOUGH ROOM
C3B4 24 2B              BHS   TRC220    NO

C3B6 1F 02              TFR   D,Y       POIN TO MAP
C3B8 EC A4              LDD   ,Y        CK FOR CURRENT ENTRY
C3BA 27 20              BEQ   TRC200    OK

C3BC 8E C62B            LDX   #FILC
C3BF BD CD1E            JSR   PSTRNG    PRINT COLLISION MSG
C3C2 6D A4              TST   ,Y        CK FOR DIRECTORY FILE
C3C4 2A 0A              BPL   TRC150    NO
C3C6 8E C63D            LDX   #DIR      PRINT DIRECTORY
C3C9 C6 09              LDB   #9        MSB LEVEMT
C3CB BD C4B6            JSR   PNUM
C3CE 20 07              BRA   TRC160    CONT ERROR MSG
C3D0 1F 21      TRC150  TFR   Y,X       POINT TO DIR #
C3D2 C6 04              LDB   #4
C3D4 BD C4EA            JSR   OUTDEC    PRINT COLLISION #
C3D7 8E C605    TRC160  LDX   #SECK
C3DA 20 2B              BRA   TRCERR    FINISH ERROR MSG

                        * CK FOR NEXT LINK
C3DC FC C111    TRC200  LDD   DIRNUM    SET IN MAP
C3DF ED A4              STD   ,Y
C3E1 8E C880    TRC220  LDX   #WRKFCB+#40
C3E4 10AE C10F          LDY   TRKSEC    GET CURRENT T/S
C3E8 EC 84              LDD   ,X
C3EA 27 0A              BEQ   TRC300    LAST SECTOR
C3EC 109F C108          STY   PRV_TS    UPDATE PREVIOUS
C3F0 FD C10F            STD   TRKSEC    NEW CURRENT TRK/SEC
C3F3 7E C324            JMP   TRCLOP    DO AGAIN

                        * CK FOR ENDING NOT MATCHING WITH DIRECTORY
C3F6 7D C111    TRC300  TST   DIRNUM
C3F9 2B 08              BMI   TRC400    BYPASS FOR DIR
C3FB 10BC C108          CMPY  END_TS
C3FF 27 05              BEQ   TRC400

C401 8E C648            LDX   #ENDERR   GIVE END ERROR
C404 8D 01              BSR   TRCERR

C406 39         TRC400  RTS

                        * PRINT TRACE ERROR MESSAGE
C407 BD CD1E    TRCERR  JSR   PSTRNG
C40A 8E C10F            LDX   #TRKSEC   GIVE FAILING TRACK/SEC
C40D BD CD45            JSR   OUTADR
C410 8E C659            LDX   #PRVSEC   PRINT PREVIOUS TRACK/SEC
C413 BD CD1E            JSR   PSTRNG
C416 8E C10D            LDX   #PRV_TS
C419 BD CD45            JSR   OUTADR
C41C 39                 RTS
```

```
                        * PRINT S.I.R.
C410 8E  C69C  PSIR     LDX     #SIRHDR
C420 BD  CD1E           JSR     PSTRNG
C423 CC  FFFF           LDD     #$FFFF      FLAG SIR MODE
C426 FD  C111           STD     DIRNUM      AND SET FOR DIR TRACE
C429 8E  0190           LDX     #$1AFC0+$40+16 POINT TO SIR DATA

                        * PRINT DISK NAME OR DIRECTORY POINTED BY X
C42C BD  CD24  PRTNAM   JSR     PCRLF
C42F 7D  C111           TST     DIRNUM      CK FOR SIR
C432 2B  0F             BMI     PN10        YES
C434 34  10             PSHS    X
C436 8E  C111           LDX     #DIRNUM
C439 C6  04             LDB     #4
C43B BD  C4EA           JSR     OUTDEC      PRINT DIRECTORY #
C43E BD  C4C7           JSR     PSPC2
C441 35  10             PULS    X

C443 C6  08     PN10    LDB     #8
C445 34  10             PSHS    X           SAVE POINTER
C447 8D  60             BSR     PNAM        PRINT NAME
C449 7D  C111           TST     DIRNUM      DIR?
C44C 2B  05             BMI     PN20
C44E 86  2E             LDA     #'.
C450 BD  CD10           JSR     PUTCHR
C453 C6  03     PN20    LDB     #3
C455 8D  5F             BSR     PNAM        PRINT EXTENSION

C457 7D  C111           TST     DIRNUM
C45A 2A  09             BPL     PN30
C45C C6  05             LDB     #5
C45E BD  C4E8           JSR     DEC5        PRINT VOL#
C461 8D  62             BSR     PSPC4
C463 20  09             BRA     PN40
C465 8D  60     PN30    BSR     PSPC2
C467 BD  CD3C           JSR     OUTHEX      PRINT PROTECTION
C46A 8D  58             BSR     PSPC2
C46C 30  02             LEAX    2,X

C46E BD  C4F4  PN40     JSR     PRTSEC      PRINT STARTING TRACK/SEC
C471 BD  C4F4           JSR     PRTSEC      ENDING T/S
C474 C6  04             LDB     #4
C476 BD  C4EA           JSR     OUTDEC      PRINT SIZE
C479 7D  C111           TST     DIRNUM
C47C 2B  0B             BMI     PN50
C47E 8D  47             BSR     PSPC2
C480 86  53             LDA     #'S         ASSUME SEQUENTIAL FILE
C482 A0  01             TST     0,X++
C484 27  02             BEQ     PN45        YES
C486 86  52             LDA     #'R         NO RANDOM
C488 8D  CD10   PN45    JSR     PUTCHR

C48B 8D  3C     PN50    BSR     PSPC
                        * PRINT DATE
C48D 8D  3F             BSR     PNUM        MONTH
C48F 86  2D             LDA     #'-
C491 8D  CD10           JSR     PUTCHR
C494 8D  38             BSR     PNUM        DAY
C496 86  2D             LDA     #'-
C498 8D  CD10           JSR     PUTCHR
C49B 8D  31             BSR     PNUM        YEAR

C49D 7D  C111           TST     DIRNUM
C4A0 2A  04             BPL     PN55        NOT SIR
C4A2 8D  23             BSR     PSPC2
C4A4 8D  4E             BSR     PRTSEC      GIVE MAX T/S

C4A6 35  10     PN55    PULS    X           CK FOR DELETE
C4A8 A0  84             TST     ,X
C4AA 2A  07             BPL     PN60        NO
C4AC 8E  CA07           LDX     #DEL
C4AF C6  07             LDB     #7
C4B1 8D  03             BSR     PNAM
C4B3 7E  CD24   PN60    JMP     PCRLF

                        * PRINT STRING POINTED BY X FOR B CHAR
C4B6 A6  80     PNAM    LDA     ,X+
C4B8 2B  02             BMI     PNAM5       DELETED NAME
C4BA 26  02             BNE     PNAM10      NULL
C4BC 86  20     PNAM5   LDA     #$20
C4BE BD  CD10   PNAM10  JSR     PUTCHR

C4C1 5A                 DECB
C4C2 26  F2             BNE     PNAM
C4C4 39                 RTS

C4C5 8D  00     PSPC4   BSR     PSPC2
C4C7 8D  00     PSPC2   BSR     PSPC
C4C9 86  20     PSPC    LDA     #$20
C4CB 7E  CD10           JMP     PUTCHR

                        #PRINT 2 CHAR DEC NUMBER POINTED BY X
C4CE E6  80     PNUM    LDB     ,X+         GET NUM
C4D0 4F                 CLRA                COUNTER

C4D1 C0  0A     PNUM10  SUBB    #10
C4D3 25  03             BCS     PNUM20
C4D5 4C                 INCA
C4D6 20  F9             BRA     PNUM10
C4D8 CB  0A     PNUM20  ADDB    #10         RESTORE
C4DA 34  04             PSHS    B           SAVE REMAINDER
C4DC 8D  30             ADDA    #'0         MAKE NUMBER
C4DE 8D  CD10           JSR     PUTCHR
C4E1 35  02             PULS    A           GET REMAINDER
C4E3 8D  30             ADDA    #'0
C4E5 7E  CD10           JMP     PUTCHR


                        * FIX FOR FLEX OUTDEC
C4E8 C6  05     DEC5    LDB     #5          PRINT 5 CHAR
C4EA 34  10     OUTDEC  PSHS    X
C4EC 8D  CD39           JSR     FLIDEC
C4EF 35  10             PULS    X
C4F1 30  02             LEAX    2,X
C4F3 39                 RTS

                        * PRINT TRACK/SECTOR POINTED BY X
C4F4 8D  CD3C   PRTSEC  JSR     OUTHEX      PRINT TRACK
C4F7 30  01             LEAX    1,X
C4F9 86  2F             LDA     #'/
C4FB 8D  CD10           JSR     PUTCHR
C4FE 8D  CD3C           JSR     OUTHEX      SECTOR
C501 30  01             LEAX    1,X
C503 20  C4             BRA     PSPC

C505 8D  CD1E   ERR     JSR     PSTRNG      PRINT ERROR MESSAGE
C508 7E  CD03           JMP     WARMS       AND EXIT

C508 B6  FF     PROMPT  LDAA    #$FF        POINT TO CRT
C50B B7  CC22           STA     OSWTCH
C510 8D  CD1E           JSR     PSTRNG      PRINT REQUEST
C513 8D  CD15           JSR     GETCHR      GET RESPONSE
C516 84  5F             ANDA    #$5F        FORCE UPPERCASE
C518 7F  CC22           CLR     OSWTCH      POINT TO PRINTER
C51B 81  59             CMPA    #'Y         CK FOR YES
C51D 39                 RTS

C51E 42 41 44 20  IOERR FCC     'BAD IOR $',4
C528 42 41 44 20  BSIR  FCC     'BAD SIR'
C52F 2B 20 46 41  FATAL FCC     '- FATAL !',4
C539 4B 41 58 20  TS_MSG FCC    'MAX TRK/SEC '
C545 4F 4B 20 59  OK    FCC     'OK(Y/N)',4
C54D 4C 4B 54 27  CLRMSG FCC    'LNT'D MEMORY-ALL SECTORS'
C565 20 4E 4F 54          FCC   ' NOT CK'D',$0D,$0A
C570 2A 2A 20 44  CONSOL FCC    '** DON'T CONSOLIDATE FREE CHAIN **',4
C593 50 52 49 4E  SECMAP FCC    'PRINT EACH LINK Y/N?',4
C5A8 46 52 45 45  FCHAR FCC     'FREE SECTORS-IN HAP-IN SIR',4
C5C3 43 4F 4E 53  FCPRMT FCC    'CONSOLIDATE FREE CHAIN Y/N?',4
C5D6 46 41 54 41  FDIR  FCC     'FATAL DIR ERR',4
C5E8 49 4E 56 41  BADSEC FCC    'INVALID TRK/SEC',4
C5F8 52 44 20 45  READSEC FCC   'RD ERR-',4
C605 41 54 20 53  SECN  FCC     'AT SECTOR-',4
C610 52 45 43 4F  RECN  FCC     'RECORD #ERR '
C61C 57 41 53 2F          FCC   'WAS/SHOULD BE ',4
C62B 43 4F 4C 4C  FILC  FCC     'COLLISION WITH # ',4
C63B 44 49 52 45  DIR   FCC     'DIRECTORY-',4
C648 45 4E 44 20  ENDERR FCC    'END MISMATCH AT-',4
C659 50 52 56 53  PRVSEC FCC    'PRVS TRK SEC-',4
C667 46 49 4C 45  DIRHDR FCC    'FILE#      NAME     PRT  DE#'
C683 20 45 4E 44          FCC   ' END     SIZE TYPE  DATE',4
C69C 44 49 53 4B  SIRHDR FCC    'DISK NAME   VOL'
C6AB 20 20 20 20          FCC   '     F.C.START END  SIZE '
```

The program I am offering is named "PRICE". Anyone who is in business today must, at some time, build a price schedule. All price schedules are tedious, regardless of their makeup. This program is an effort to remove some of the tedium, improve the accuracy and insert some uniformity into the building of price schedules. It is written in T.S.C. XBASIC and operates under FLEX 9. It is being submitted to you on S.S.S.D. 5¼ inch disc and a printed copy. Also a printout of the end result showing headings, part numbers and columns.

It can be modified to fit most small businesses. Column headings can be modified in lines 2020, 2040 and 2060. Percent of profit is calculated and can be modified in lines 610, 710, 810, 910, etc. Line 1610 prints zeroes. If a typing error creeps in there is error recovery routine so that the last line typed can be done over without starting at the top.

To use the program one must enter the following on the keyboard:

QUESTIONS ASKED:	ENTER:
IS THIS ERROR RECOVERY (Y OR N)	N
ENTER COMPANY NAME	DELUXE BATTERY CO.
ENTER PART NUMBER	1212
ENTER DESCRIPTION	FLASHLIGHT B TTERY
ENTER COST OF PRODUCT	.49
ENTER CODE NUMBER FOR O.T.C. (1 TO 11)	7
ENTER CODE NUKBER FOR 1 TO 5	5
ENTER CODE NUMBER FOR 6 TO 11	3
ENTER CODE NUMBER FOR 12 OR CARTON	2
ARE THE ENTRIES FOR THIS PART NUMBER BREAK CORRECT (Y OR N)	Y

At this point the data goes to the printer and the program branches back so the next part number and data can be entered.

Thanks again, Don, for giving we readers the opportunity to spread the word.

Sincerely,

James W. Ivers

James W. Ivers

```
30 REM   : PRICE : BY JAMES W. IVERS
60 REM                4021 BOTH ST.
70 REM                DES MOINES, IOWA  50322
90 GOSUB  3500
95 OPEN "O",#1,"PRICE.BAS" AS 0
97 PRINT "A PROGRAM TO CALCULATE AND PRINT UP TO FOUR SELLING PRICES"
99 PRINT " FROM THE COST PRICE"
100 PRINT:PRINT
102 INPUT "HOW MANY PRICE COLUMNS DO YOU WANT?  1 TO 4 ";C%
104 IF C%=2 THEN L%=3
106 IF C%=3 THEN L%=5
108 IF C%=4 THEN L%=7
110 IF C%=1 THEN GOSUB 3700
112 IF C%>4 THEN GOSUB 3700
114 GOSUB 3300
116 PRINT "IS THIS ERROR RECOVERY ?  Y OR N ";:INPUT#(INCH$(0)
118 PRINT
120 IF TMP$="N" THEN 200
    GOSUB 3500
125 GOSUB 3500
130 PRINT: INPUT "ENTER COMPANY NAME ";A$
140 GOSUB 2000
145 PRINT #0,TAB(45);DATE$
150 GOSUB 3000
155 PRINT
160 GOSUB 2020
170 GOSUB 3400
185 PRINT
190 GOSUB 3500
200 INPUT "ENTER PART NO....OR (STOP) TO END ";P1$
205 PRINT:PRINT
210 IF P1$="STOP" THEN 1900
220 INPUT "ENTER DISCRIPTION ";P2$
225 PRINT:PRINT
230 INPUT "ENTER COST OF PRODUCT";I
235 FOR J=1% TO 4
240 GOSUB 3500
260 PRINT TAB(5);"1 - 20%";TAB(50);"6 - 33.3"
280 PRINT TAB(5);"2 - 22.2";TAB(50);"7 - 36%"
300 PRINT TAB(5);"3 - 25%";TAB(50);"8 - 40%"
320 PRINT TAB(5);"4 - 27.5%";TAB(50);"9 - 45%"
340 PRINT TAB(5);"5 - 30%";TAB(50);"10 - 53%"
345 PRINT:PRINT TAB(15);"11 - WILL PRINT  (0.00)  HIT RETURN TWICE"
360 PRINT:PRINT
380 IF J=2 THEN 480
400 IF J=3 THEN 520
420 IF J=4 THEN 560
440 INPUT "ENTER CODE NUMBER  FOR O.T.C. (1 TO 11)";B
450 IF B >11 THEN 3600
460 ON B  GOTO 600,700,800,900,1000,1100,1200,1300,1400,1500,1600
480 INPUT "ENTER CODE NUMBER FOR 1 TO 5 ";B
490 IF B>11 THE  GOTO 3600
500 ON B GOTO 600,700,800,900,1000,1100,1200,1300,1400,1500,1600
520 INPUT "ENTER CODE NUMBER FOR  6 TO 11 ";B
530 IF B>11 THEN GOTO 3600
535 ON B GOTO 600,700,800,900,1000,1100,1200,1300,1400,1500,1600
540 INPUT "ENTER CODE NUMBER FOR 12 OR CTN. ";B
550 IF B>11 THEN GOTO 3600
560 ON B GOTO 600,700,800,900,1000,1100,1200,1300,1400,1500,1600
600 A=I/.8
610 A1=INT(A*100+.5)/100
620 IF J=1 THEN D1=A1
630 IF J=2 THEN M1=A1
640 IF J=3 THEN M2=A1
650 IF J=4 THEN M3=A1
670 NEXT J
680 GOTO 1 00
700 B=I/.775
710 B1=INT(B*100+.5)/100
720 IF J=1 THEN D1 B1
730 IF J=2 THEN M1=B1
740 IF J=3 THEN M2=B1
750 IF J=4 THEN M3=B1
760 NEXT J
780 GOTO 1200
800 C=I/.75
810 C1=INT(C*100+.5)/100
820 IF J=1 THEN D1=C1
830 IF J=2 THEN M1=C1
840 IF J=3 THEN M2=C1
850 IF J=4 THEN M3=C1
860 NEXT J
880 GOTO 1700
900 D=I/.725
910 D1=INT(D*100+.5)/100
920 IF J=1 THEN D1=D1
930 IF J=2 THEN M1=D1
940 IF J=3 THEN M2=D1
950 IF J=4 THEN M3=D1
960 NEXT J
980 GOTO 1700
1000  E=I/.7
1010 E1=IN (E*100+.5)/100
1020 IF J=1 THEN D1=E1
1030 IF J=2 THEN M1=E1
1040 IF J=3 THEN M2=E1
1050 IF J=4 THEN M3=E1
1070 NEXT J
1080 GOTO 1700
1100 F=I/.2
1110 F1=INT(F*100+.5)/100
1120 IF J=1 THEN D1=F1
1130 IF J=2 THEN M1=F1
1140 IF J=3 THEN M2=F1
1150 IF J=4 THEN M3=F1
1170 NEXT J
1180 GOTO 1700
1200 G=I/.64
1210 G1=INT(G*100+.5)/100
1220 IF J=1 THEN D1=G1
1230 IF J=2 THEN M1=G1
1240 IF J=3 THEN M2=G1
1250 IF J=4 THEN M3=G1
1270 NEXT J
1280 GOTO 1700
1300 H=I/.6
1310 H1=INT(H*100+.5)/100
1320 IF J=1 THEN D1=H1
1330 IF J=2 THEN M1=H1
1340 IF J=3 THEN M2=H1
1350 IF J=4 THEN M3=H1
1370 NEXT J
1380 GOTO 1700
1400 I=I/.55
1410 I1=INT(I*100+.5)/100
1420 IF J=1 THEN D1=I1
1430 IF J=2 THEN M1=I1
1440 IF J=3 THEN M2=I1
1450 IF J=4 THEN M3=I1
1470 NEXT J
1480 GOTO 1700
1500 L=I/.47
1510 L1=INT(L*100+.5)/100
1520 IF J=1 THEN D1=L1
1530 IF J=2 THEN M1=L1
1540 IF J=3 THEN M2=L1
1550 IF J=4 THEN M3=L1
1570 NEXT J
1580 GOTO 1700
1600 N$=INCH$(0)
1610 IF J=1 THEN D1=VAL(N$)
1620 IF J=2 THEN M1=VAL(N$)
1630 IF J=3 THEN M2=VAL(N$)
1640 IF J=4 THEN M3=VAL(N$)
1650 NEXT J
1660 GOTO 1700
1700 REM PRINT ROUTINES

1705 GOSUB 3500
1720 PRINT TAB(10);"COST AS ENTERED---->";I
1730 PRINT:PRINT
1740  PF =\      \2345367\ \2345367890123456789001\   ###.##    ###.##    ###.##    ###.##
1750 PRINT USING PF$,P1$,P2$,D1,M1,M2,M3
1760 PRINT:PRINT
1770 PRINT "ARE THE ENTRIES FOR THIS PART NO. CORRECT  (Y OR N)";:PR$=INCH$(0)
1780 IF PR$()"Y"  THEN  190
1790 PRINT #0,USING PF$,P1$,P2$,D1,M1,M2,M3
1800 GOTO  190
1900 CLOSE 0
1910 END
2000 PRINT #0,TAB(60-.5*LEN(A$));A$
2010 PRINT #0:RETURN
2020 PRINT #0,"     PART NO.";TAB(14);"DISCRIPTION";
2040 PRINT #0,TAB(39);"O.T.C.";TAB(47);"1 TO 5";
2060 PRINT #0,TAB(58);"6 TO 11";TAB(64)"12 OR CTN."
   70 RETURN
3000 PRINT #0,TAB(5);
3005 FOR V= 1 TO 69
3010 PRINT #0,"-";
3020 NEXT V
3028 PRINT #0
3030 RETURN
3500  PRINT CHR$(27)(CHR$(69);CHR$(12)
3510 RETURN
3600 PRINT,PRINT;PRINT "CODE  NUMBER OUT OF RANGE---RE-ENTER."
3610 FOR V=1 TO 1000
3620 NEXT V
3630 GOTO 240
3700 PRINT:PRINT:PRINT "NUMBER OUT OF RANGE.--RE-ENTER."
3710 FOR V=1 TO 1000
3720 NEXT V
3730 GOTO 120
```

DELUXE BATTERY CO.

09-SEP-09

PART NO.	DISCRIPTION	O.T.C.	1 TO 5	6 TO 11	12 OR CTN.
1212	FLASHLIGHT BATTERY	0.77	0.70	0.63	0.63
1366	PENLIGHT BATTERY	0.73	0.67	0.63	0.61
1598	LANTERN BATTERY	2.81	2.57	2.40	2.32
2466	FENCE BATTERY	5.03	4.60	4.29	4.15
3333	9 VOLT RADIO	1.38	1.26	1.17	1.14

FOR SALE: Smoke Signal 6809 system w/OS9-1, Basic09, Assembler Editor, Debug. 62k RAM, DCB4a, 2 ser/2 par, 2DS/DD MP152DD, less than 1 year old-$3700. H19 Term and MX80 printer-$800. Package price $4300.00.
Doug Ramers (302) 674-6386, 697-3444.

Complete 68000 system. 32k RAM, 16k ROM, parallel printer port, cassette port, auxiliary RS232 port and timer inside an H-19. Monitor includes a host of commands including a single-line assembler/dis-assembler and 'help'. Can redirect most I/O or operate H-19/aux in transparent mode. Excellent condition with all manuals only $1750.
Dick Riggs, 2865 Akron, East Point, Ga 30344.

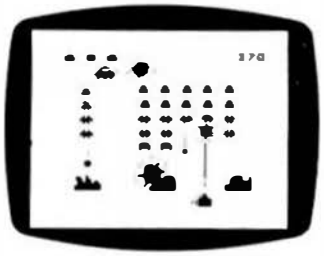
FOR SALE: BFD 5 inch disk controller with SWTPC 6809 CPU. Runs FLEX or SSB at $8000 or $E000. APF Imagination 6800 color computer with ROM Basic, joysticks and Basic Tutor. Other 6800/09 boards.
Call John (216)747-6781 anytime.

FOR SALE: SWTPC AC-30 cassette interface complete with audio cables. And the original MP-AI 6800 processor board (purchased in October 1976,a real antique!) Both in mint working condition. The first reasonable offer takes either or both.
Dwight Lanpher, Box 472, N E Harbor, ME 04662.

WANTED: TSC Minitlex DOS and Disc at $7000-7FFF. Also SWTPC 8k memory board, MP8M.
Steve Stanley, 1 Whitney Rd, Latham, N Y 12110
(518) 785-5089.

TRS 80 **COLOR COMPUTER** TRS 80

SPECTRAL ASSOCIATES, the LEADER in TRS-80 Color Computer software, presents the best in 16K machine language, fast action, high resolution arcade games. Extended Basic not required.

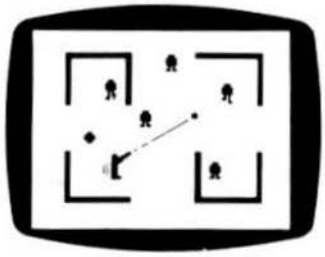


COSMIC INVADERS

The best invaders type game available. Complete with 16 skill levels, dynamite sound and 4 color hi-res graphics. Can you dodge the bombs long enough to blast the alien ship and invaders? $21.95

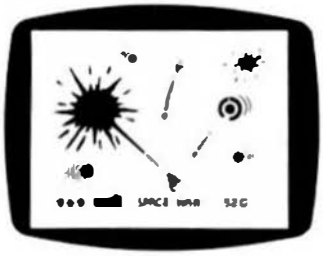
GHOST GOBBLER

An exciting version of the popular PAC-MAN game with power dots, 8 bonus shapes, super sound and 16 skill levels. Hours and hours of challenging fun. Joysticks $21.95



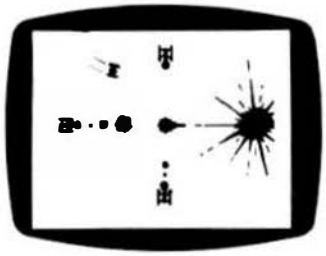
ROBOT BATTLE

Can you dodge the never ending horde of robots while avoiding the fatal touch of the Android? Realistic voices and 16 skill levels provide a tremendous arcade type challenge. Joysticks, $21.95.



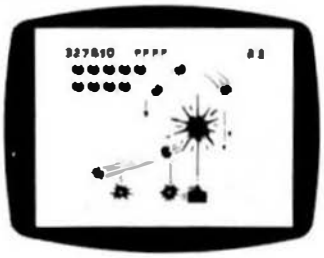
SPACE WAR

Maneuver your spaceship through the defenses of the Death Star for the killing shot while avoiding space mines, enemy ships and the gravity pull of the BLACK HOLE. Two color hi-res graphics - $21.95 Joysticks



COLOR ZAP

Super new space game! You must dodge the incoming photon torpedos before they can penetrate the defenses of your ship. Once you think you've got the upper hand, defensive fighters may appear and attack. $9.95



GALAX ATTAX

If you like Space Invaders, you'll love Galax Attax! Alien fighters leave formation to attack your ground base and you must fight them off! Exciting, fast-action machine language game with super sound and high resolution graphics. 16K and joysticks required. $21.95

NEW 16K MACHINE LANGUAGE GAMES

- KEYS OF THE WIZARD $19.95
  Best new adventure game available - over 200 rooms filled with creatures, tricks, treasures, magic spells. Cassette save feature built in.
- COLOR OUT $9.95
  Like Breakout, you must knock out six layers of blocks using your paddle to prevent missiles from escaping. Joysticks optional.

SUPERCHARGER

If you have a 32K E series printed circuit board, the SUPERCHARGER will let you switch your computer to full 64K RAM mode. It just plugs into the ROM PAK port and you can use BASIC or not as you wish. NO MODIFICATION is needed and it will not void your warranty. It allows you to still plug ROM pack and/or the disk controller board into the computer. ONLY $39.95

FLEXPLUS DOS

FLEXPLUS is a powerful, easy to use disk operating system. Spectral Associates has adapted TSC's FLEX to the best DOS completely compatible with Radio Shack software for use on the Color Computer. Eliminate the need for Radio Shack's TRS DOS. Use FLEXPLUS with Editor/Assembler and have the options of a full range of utilities. FLEXPLUS works on the 32K Radio Shack disk system with 64K memory chips with a High Resolution multi-screen format that supports a 24 line by 51 character display! Also included are special enhancements to Radio Shack's Disk system when you are running FLEX with single or double sided, single or double density, 35, 40 and 80 track drives.

Advantages of FLEXPLUS are:

- Best price anywhere
- Easy start-up — just type "RUN FLEX+"
- Allows you to save RS compatible disk files from FLEXPLUS
- All FLEX compatible software will run including INTERRUPT DRIVEN SOFTWARE
- NO HARDWARE MODIFICATIONS NEEDED
- Warranty will not be voided — no need to open computer
- Wide range of available software
- Requires Supercharger board
- $149.95 for FLEXPLUS, Editor/Assembler and Supercharger

THE FACTS

At last, a complete description of the "guts" of the Color Computer. Specs on all the ICs, complete schematics, theory of operation and programming examples.

$14.95

UTILITIES

- EDITOR/ASSEMBLER $34.95
- SUPER MONITOR 19.95
- EPROM PROGRAMMER 89.95
  (Program your own ROMs for the ROM PAC port)
- MAGIC BOX 24.95
  (Load MOD VIII Basic Tapes into color computer)

SPECTRAL ASSOCIATES
139 Harvard Ave.
Tacoma, Washington 98466
(206) 565-8483

WRITE FOR COMPLETE CATALOG

ADD 3% FOR SHIPPING NO COD

VISA OR MASTERCHARGE ACCEPTED

All prices in U.S. funds

ORDERS ONLY TOLL FREE

800-426-1830

EXCEPT WA, HI, AK

BUY DIRECT AND STUFF IT!

$19.50 EACH • 30 Pin Extender Board

• Dual Serial I/O Board (30-pin)

$29.50 EACH • 50 Pin Extender Board

All Extender boards fully labeled and shielded to prevent RF Interference.

$39.50 EACH

- 6809 CPU Board Baud Rate Generator
- Multiple I/O Board 2 ACIA and 2 PIA
- Video Display Board 6845 Based 80x24 Display
- Video Color Graphics Board 6847 Full Graph
- 5¼" Disc Controller Board (50 Pin)
- Modem Board Bell 103 Compatible, No hard to get parts!

$49.50 • 64K Dynamic Ram Board

- Completely Solder Masked Both Sides
- Full Documentation
- Reliable Phase I Refresh

$69.50 • Mother Board

- .093 Thick
- Baud Rate Generator on Board
- Full Address Decoding
- 4 or 16 Address Per I/O Slot

- All Boards Available Fully Stuffed
- VISA & Master Card Accepted
- Add $3.50 for shipping & handling
- All Boards Solder Masked Both Sides
- All Silk Screened Nomenclature
- Full Documentation Included

You're Right

We can't sell boards at this price for very long. But what better way to introduce you to DATA SYSTEMS "68". Order before August 31, 1982 (call 305-425-6800 for credit card orders) and realize savings of up to 50%. We guarantee your satisfaction.

DS 68

Data Systems "68"
2316 Diversified Way
Orlando, Florida 32804
(305) 425-6800

Limited Warranty

All DS68 products are guaranteed for a period of 90 days to be free of defects in materials and workmanship or return to Data Systems "68" for free replacement.

TERMS: Check, MasterCard, Visa. Sorry, NO C.O.D.'s. Florida residents add 5½ sales tax. Specify type board(s) and quantity of each when ordering.

TEN MOST-ASKED QUESTIONS

ABOUT DYNACALC™

THE ELECTRONIC SPREAD-SHEET FOR 6809 COMPUTERS

1. **What is an electronic spread-sheet, anyway?**
   Business people use spread-sheets to organize columns and rows of figures. DYNACALC simulates the operation of a spread-sheet without the mess of paper and pencil. Of course, corrections and changes are a snap. Changing any entered value causes the whole spread-sheet to be re-calculated based on the new constants. This means that you can play, 'what if?' to your heart's content.

2. **Is DYNACALC just for accountants, then?**
   Not at all. DYNACALC can bs used for just about any type of job. Not only numbers, but alphanumeric messages can be handled. Engineers and other technical users will love DYNACALC's sixteen-digit math and built-in scientific functions. There's even a built-in sort command, so you could use DYNACALC to manage small data bases - up to 256 records.

3. **What will DYNACALC do for ME?**
   That's a good question. Basically the answer is that DYNACALC will let your computer do just about anything you can imagine. Ask your friends who have VisiCalc, or a similar program, just how useful an electronic spread-sheet program can be for all types of household, business, engineering, and scientific applications.

4. **Do I have to learn computer programming?**
   NO! DYNACALC is designed to be used by non-programmers, but even a Ph.D. in Computer Science can understand it. Built-in HELP messages are provided for quick reference to operating instructions.

5. **Do I have to modify my system to use DYNACALC?**
   Nope. DYNACALC uses any standard 6809 configuration, so you don't have to spend money on another CPU board or waste time learning another operating system.

6. **Will DYNACALC read my existing data files?**
   You bet! DYNACALC has a beautifully simple method of reading and writing data files, so you can communicate both ways with other programs on your system, such as the Text Editor, Text Processor, Sort/Merge, RMS data base system, or other programs written in BASIC, C, PASCAL, FORTRAN, and so on.

7. **How fast is DYNACALC?**
   Very. Except for a few seldom-used commands, DYNACALC is memory-resident, so there is little disk I/O to slow things down. The whole data array (worksheet) is in memory, so access to any point is instantaneous. DYNACALC is 100% 6809 machine code for blistering speed.

8. **Is there a version of DYNACALC for MY system?**
   Probably. You need a 6809 computer (32k minimum) with FLEX or UniFLEX operating system. A version for OS-9 is also in the works. You also need a decent CRT terminal, one with at least 80 characters per line, and direct cursor addressing. If your terminal isn't smart enough for DYNACALC, you probably need a new one anyway. The UniFLEX version of DYNACALC also allows you to mix different brands of terminal on the same system. There's also a special version of DYNACALC for Color Computers equipped with FLEX and Data-Comp's F-MATE. A version for Frank Hogg's Color Computer FLEX is also being done.
9. **How much does DYNACALC cost?**
   The FLEX versions are just $200 per copy; UniFLEX version $395. Foreign orders add $10 per copy for postage. We encourage dealers to handle DYNACALC, since it's a product that sells instantly upon demonstration. Call or write on your company letterhead for more information.
10. **Where do I order DYNACALC?**
    See your local DYNACALC dealer, or order directly from CSC at the address below. We accept telephone orders from 10 a.m. to 6 p.m., Monday through Friday. Call us at 314-576-5020. Your VISA or MasterCard is welcome. Please specify diskette size for FLEX versions. Software serial number is required for the UniFLEX version of DYNACALC.

ORDER YOUR DYNACALC™ TODAY

Foreign Dealers:

Australia & Southeast Asia: order from Paris Radio Electronics, 7A Burton St., Darlinghurst, NSW 2010 Sydney. Telephone: 02-357-5111.

United Kingdom: order from Compusense, Ltd., PO Box 169, London N13 4HT. Telephone: 01-882-0681.

Scandinavia: order from Swedish Electronics hk AB, Murargatan 23-25, Uppsala S-754 37 Sweden. Telephone: 18-25-30-00.

Computer Systems Center

13461 Olive Blvd.

Chesterfield, MO 63017

(314) 576-5020

UniFLEX software prices include maintenance for the first year.

DYNACALC, DYNAMITE, and DYNAMITE + are trademarks of Computer Systems Center.

F-MATE is a trademark of Data-Comp.
VisiCalc is a trademark of VisiCorp.
OS-9 is a trademark of Microwara and Motorola.
FLEX and UniFLEX are trademarks of TSC.

ALSO FROM CSC

DYNAMITE +

"THE CODE BUSTER"

now available for UniFLEX
OS-9 version soon

DYNAMITE + is a new version of DYNAMITE, our popular 6809/6800 disassembler package for 6809 FLEX. Present users of DYNAMITE can upgrade to DYNAMITE + by sending us the original DYNAMITE diskette and $40 (plus $5 for foreign postage). DYNAMITE + does everything DYNAMITE does, and more! A cross-reference generator has been added, label files are now maintained only in text form (LABEL EQU $xxxx), and boundary file specifications have been tremendously simplified, which makes it easier to disassemble large programs containing lots of big tables.

The UniFLEX version of DYNAMITE + does everything the FLEX version does, and also automatically handles system calls and 'info' areas.

DYNAMITE + is available for $100 per copy on FLEX (specify diskette size), and $300 on UniFLEX. Foreign orders add $5 per copy for postage.

OmegaSoft Puts Pascal Power In Program Development

Finally. A fast, efficient way to design, code, debug and maintain your microprocessor software. OmegaSoft's Pascal. The software development tool that takes the sting out of programming. For industrial or business use, large or small scale.

How does it work? Simple. We've compiled Pascal into the compact native code that is most common to professional applications. OmegaSoft Pascal is designed to integrate easily with existing software development tools available on most 6809 operating systems.

Single Pass Punch

Our compact single-pass compiler quickly translates Pascal into 6809 assembly language code. This means less time-wasting disk swapping and fewer multiple passes, with greater success rates. OmegaSoft Pascal is based on the proposed ISO standard with compatible extensions designed to make Pascal an important tool in industry and business. Byte wide variables allow convenient access to I/O devices. Custom I/O devices can be used in place of normal Pascal INPUT and OUTPUT files.

Practical Yet Precise

OmegaSoft's Pascal long, 32-bit integer support also allows representation of money amounts without speed penalties of BCD implementations. And our 7 digit real format is compatible with AM9511 APU with options to use this chip instead of software routines.

OmegaSoft Pascal's random access file is excellent for fast access of data for business applications programs and other mass storage uses. Sophisticated applications programs are simplified with dynamic length strings, such as:

- Process control.
- Communication systems.
- Robotics.
- Automatic test systems.
- Microcode generation.
- Scientific computation.
- Development system software.

High Speed, High Performance

OmegaSoft Pascal generates high speed code without sacrificing ROMability, position independence or reentrancy. Modular compilation is supported along with several methods of interfacing to assembly language variables and routines, including interrupt procedures.

OmegaSoft Pascal's runtime library provides that only the modules necessary to execute the Pascal program are loaded, which translates into much smaller object modules and reduced memory costs.

Our symbolic debugger interfaces with the assembly language output of the compiler for fast, accurate program execution. Debugger commands allow other exciting possibilities:

- Displaying and changing variables using Pascal names.
- Tracing through statements.
- Setting breakpoints at the start of Pascal statements.

Features

- Compatible with MDOS™ and XDOS™, FLEX™, DOS69™, and OS-9™.
- Entry, External, and Absolute attributes for variables and procedures.
- Optimized in-line code generated wherever possible rather than subroutine calls.
- Compiler generates 6809 assembly language – can either be used with the debugger or assembled and linked to form a loadable object module.
- Standard data types plus long-integer, hex, dynamic length string, and device.
- Expanded operations for pointers and characters allows low level code to be written in Pascal.
- Extensive type conversion functions.
- Compiler – Requires 48K system.
- Symbolic Debugger to run on any of the operating systems.
- Utilities to assist in creating a "CHAIN" file for automatically linking the user program with the runtime library.
- Manuals – detailed language reference handbook and a configuration manual.
- Runtime library and source.
- ISO validation report available.
- Debugger allows display and change of variables using their symbolic names and stepping through the program one statement at a time.

Single unit prices start at $425 (domestic).

Support products include: OmegaSoft's Motorola compatible Relocatable Assembler/Linker and for higher speed OmegaSoft's Arithmetic Processor option uses the AMD9511 to support integer, longinteger, and real calculations for Pascal runtime. Available soon will be a multitasking kernel and a high speed translator between compiler output and relocatable object code.

TM: MDOS, XDOS, Motorola trademarks; Flex, TSC trademark; DOS69, Smoke Signal Broadcasting trademark; OS-9 Microware trademark.

OmegaSoft
P.O. Box 70265
Sunnyvale, CA 94086
Telephone (408) 733-6979

INFOMAG DBMS

A data base management system specifically designed for microprocessor based computer systems.

- You specify new databases and screen entry masks.
- Define and work with subgroups of the master database.
- Sort groups.
- Audit trail can be maintained.
- Password protection.
- Control code definition for different terminals and printers.
- Columnar reports, modular reports, reports drawing from primary and secondary files.
- Statistics reports.
- Database files can be accessed by userwritten BASIC programs.

FLEX - $295.00 UniFLEX -$395.00

OSBORNE BUSINESS PROGRAMS

This ENHANCED IMPLEMENTATION of the Osborne and Associates Business Programs is the only implementation available with the full capability of the original Wang Minicomputer version.

FEATURES INCLUDE:

- **KEYED FILES** to eliminate slow searches and sorts.
- **PASSWORD** and **MASTER PASSWORD PROTECTION** to limit unauthorized access to your business data.
- **LINKED** can be run linked to G/L.

ACCOUNTS RECEIVABLE This package is an open invoice system, it prints aging reports, monthly statements, open item listings, etc. You can maintain trade discount schedules and scheduled penalties for late payment. FLEX.....$295 UniFLEX.....$395

ACCOUNTS PAYABLE This package is an invoice-linked system, it prints aging reports and will print out your checks. FLEX.....$295 UniFLEX.....$395

GENERAL LEDGER Accepts postings to the various accounts from external sources: Accounts Payable, Accounts Receivable, and Cash Journal. The normal posting is double entry to reduce off-balance posting errors. FLEX.....$295 UniFLEX.....$395

6809 SOFTWARE TOOLS

6800 RRMAC (Relocatable Recursive Macro Assemblers)
These are true Macro assemblers with external linkage capability. Variables may be local, global, or external. One assembler produces relocateable 6800 code which is loaded with the included Linking Loader. The second assembler, a cross assembler runs on your 6800 and produces relocatable 6809 object code from existing 6800 or new 6809 source files. Two assemblers plus a Linking Loader are including in this package.

FLEX.....$150

6809 RRMAC (Relocatable Recursive Macro Assemblers)
One assembler produces relocatable 6809 code which is loaded with the included link editor. The second 6809 assembler, a cross assembler, runs on your 6809 and produces relocatable 6800 code.

FLEX.....$150

SUPER SLEUTH DISASSEMBLER, enables the user to examine and/or modify object program files on disk or in memory on 6800/1/9 systems.

FLEX.....$99

GREAT PLAINS COMPUTER CO.

P.O. BOX 916 / IDAHO FALLS, ID 83402 / PH: (208) 529-3210
Visa and MasterCard accepted.

HOME CONTROL

HCOR software provides TOTAL control of ALL 256 possible devices. We provide the software and an adapter cable, you provide the RADIO SHACK PLUG N' POWER(tm) controller (026-1102) and a parallel port. This real time software may be adapted for whatever CPU speed you may be running and allows parameters to be modified for noisy electrical environments. Remember that NO ultrasonic links are required. Two NEW commands have been added to the standard six commands allowing you to LOCK and UNLOCK all devices under your control.
tSAVAILABLE FOR PERCOR AND STANDARD OS9 FORMAT ROM!!
REQUIRES APPROXIMATELY 1K TO RUN

5" DISK WITH ADAPTER & DOCUMENTATION-------$35.00

DP8 DUAL PORT SERIAL BOARD

COMPATIBLE WITH BOTH 6830 & 6830-C
FULL RTS, DCD, CTS etc. JUMPER CONTROL
ALL CHIPS SOCKETED--USES 1488/1489 DRIVERS
ALL SUPPLY VOLTAGES REGULATED (+5, +12, -12)
TWO DB25 CONNECTORS ON BOARD FOR EASY HOOKUP

COMPLETE & TESTED WITH DOCUMENTATION-------$69.00

TG-1 BIT RATE GENERATOR

9 STANDARD BIT RATES FROM 110-19.2K SIMULTANEOUSLY
USES NO PARTS FROM YOUR PRESENT SYSTEM
FACILITATES UPGRADING TO 6850C STATUS

COMPLETE & TESTED WITH DOCUMENTATION-------$49.00

GOLD PLATED CONNECTORS ARE STANDARD

PLUG N' POWER is a trademark of RADIO SHACK.
OS9 is a trademark of MICROWARE SYSTEMS CORP.
SLEUTH is a trademark of SILICA INC.

TEXAS COMPUTER

add for shipping $3.00

P.O. BOX 120818 ARLINGTON, TX 76012

VISA/MC 817-275-1848

EPROM PROGRAMMER

Shown assembled. EPROM not included.

For single supply 2516, 2716 & 2758 EPROMs. Connects through a user supplied interface to any computer system. Interfacing requires two 8-bit ports plus hand-shake lines. One of the ports must be software controllable for input or output. Timing is done via hardware, thus is independent of MPU clock rate. Verify erased. Program — entire or partial. Auto verify after programming. Transfer contents to RAM for modifying or duplicating.

Select Documentation for:	Interface to:
6502	6820 PIA or 6522 VIA
6800	6820 PIA
6809	6820 PIA
8080/8085/Z80	8255 PPI

Comprehensive documentation booklet contains schematic, instructions for construction, check-out and use, and a well commented assembly listing for the specified MPU.

Complete kit of parts (includes ZIF socket) ..... $ 45.00
Bare PC board and Documentation ..... $ 25.00
Software listings for additional MPUs (with purchase of kit or PC board) ..... $ 5.00

Ordering: Specify MPU. Add 5% for P&H. Overseas add 10%. Ariz. residents add 5% tax.

Micro Technical Products, Inc.
814 W. Keating Ave., Dept. J
Mesa, Arizona 85202 • 602-839-8902

ALFORD'S GOOD NEWS!

UNLIMITED COMPUTER SPEECH...

Our speech synthesizers allow your computer to talk, LITERALLY! Using our VS-1 (for the 68-30 bus) or the SP-1 (for the COLOR COMPUTER), your programs can now communicate with aSpeCh! You can add a new dimension to almost any application with our synthesizers. Business programs, computer assisted instruction, industrial control, alarm systems, and anything else can benefit from the power of speech.

We don't just sell you a bunch of hardware, either. Our synthesizers come with well-documented manuals detailing speech theory, programming examples, sample dictionaries, coding tables and more. We even include software to help to integrate these products into your systems! Our unique "VOX-EDITOR" speech editing program, for instance, allows you to create and edit speech messages simply and easily. You can save the speech you created into files which can be directly read by Basic or other programs.

Both units come completely assembled & tested. No electronics knowledge or assembly is required. The SP-1 plugs directly into the ROM-PACK slot on the COLOR COMPUTER. Audio is fed back into the computer and routed to your television monitor. The VS-1 occupies one slot on the 30-pin I/O bus, and only requires the attachment of an external speaker (not supplied) for operation.

The VS-1 software is available for SSB DOS68 and all versions of 6800 FLEX (except mini-FLEX), and for SSB DOS69, FLEX-09 and OS-9, and may be ordered on 5- or 8-inch disk. The SP-1 software comes on a stable cassette (disk versions not available at this time).

TEXT TO SPEECH TRANSLATOR...

We are now offering a speech translator for the VS-1 and SP-1 synthesizers.

The SC-1/VS-1 translator allows you to send to speech files from the keyboard, or from standard disk text files. The resulting file may be "fine-tuned" with the VOX-EDITOR, thus greatly reducing the amount of time needed to generate custom messages. Also, the SC-1/VS-1 translator allows you to simply speak Your text files, allowing you to listen to your work while you do something else, eliminating the need to keep your eyes "glued" to the screen. You may also type from the keyboard directly to the translator.

The SC-1/SP-1 translator allows you to operate directly from Color Basic by adding a new port to Basic. Now, by typing "PRINT #1, "THIS IS A TEST", your computer will say "THIS IS A TEST"! By using control codes (PRINT #1,CHR$(nn)), you can make the translator pronounce everything which goes to the screen!

Flags are included in both versions which will cause the translator to pronounce punctuations, in this way, you can listen to every "dot and tittle" as it is read!

We plan to offer regular updates at a reduced price, so that as we continue to improve the translational accuracy of this product, you can take advantage of the updates without having to pay the full price!

The SC-1/VS-1 version is available for 6800 SSB DOS68, versions 5.1 and 6.0, and for all versions of FLEX (except mini-FLEX). 6809 versions are available for DOS69, FLEX-09 and OS-9. All may be ordered on 5-inch or 8-inch disk. The Color Computer version requires 16k or XRAM memory (Extended Color Basic not required), and is supplied on cassette only.

The disk versions of the SC-1 translator include the source files, so you can make custom modifications to allow you to use it with other speech boards (we admit they exist!), or to change I/O addresses, etc... The 6809 disk version is self-relocating to place it at the top of available memory and out of the way of your other programs!

PUBLIC NOTICE...

FLEX is a trademark of Technical Systems Consultants, Inc. OS-9 is a trademark of Microware Systems Corporation. Radio Shack and TRS-80 Color Computer are trademarks of the Tandy Corporation. SSB ain't so picky! (love you, Rich, Larry, Jim, Debbie, Ken, etc...!), SCREDITOR III (no, Ed, we aren't gonna change it!) is a trademark of Alford & Associates, but then who else would use the name for that matter, who can pronounce it?

TREK...

Some of you might not know it, but many of the games which you have seen written in Basic are emulations of games which were developed with compilers or assemblers for large machines. Some are pretty engrossing games, but all share two very bad problems ... question-and-answer play, and low speed. We want in the other direction. We took one of the most popular Basic games of all time, re-wrote it completely in assembly language, using full interrupt-driven, multi-tasking structure, and came up with what we feel is one of the best games around today!

Although our TREK uses alphanumerics exclusively in its display (thus making it compatible with most every memory-mapped display around), the lack of hi-res graphics is not missed due to the edge-of-the-seat action! The enemy space destroyers use multiple battle plans as they pursue you around the quadrants. Torpedoes destroy asteroids in their paths, forcing you to take continual evasive action. Damage control reports flash on and off as your ship gets and repairs are made. Energy runs down continuously, making it impossible to sit back and relax. The only way to believe that so much action could be fitted on one screen is to play it!

TREK-68/6800 is available for 6800 processors running SSB DOS68 or any version of FLEX (except mini-Flex), and may be obtained on 5- or 8-inch disk. TREK-68/6809 is available for 6809 processors running SSB DOS69 or FLEX-09, and can be shipped on 5- or 8-inch disk.

COLOR-TREK is available on cassette for the TRS-80 COLOR COMPUTER, and requires 16K or more memory (Extended Color Basic NOT required). COLOR-TREK- is identical to COLOR-TREK, except that it requires our SP-1 Sam or Push voice synthesizer for operation, as the computer uses a high speed machine code subroutine to create all program play.

TREK-68 requires a MEMORY-MAPPED display for operation as well as a source of interrupts (such as the SP-1 timer) programmed for 50-120 interrupts per second. Also, a serial or Parallel keyboard port is required (will not run with the SP-C interface).

FANTASTIC PROOFREADER...

We are now offering Peter Stark's SPELL'N FIX proofreading program for both FLEX and SSB! We think that this program is the best spelling checker/corrector available today, and is definitely the fastest and most flexible such program that we have used.

SPELL'N FIX provides a number of features that we have not seen on any other such program ... features such as multiple dictionaries, compressed dictionaries (35000 words on a single 5-inch diskette!), automatic dictionary updates, the option of simply printing errors, marking them in the file being proofread, or correcting them! Sector limits may be set to limit the length of file on each disk, text files may be split up, the dictionary split, and best of all, SPELL'N FIX doesn't let you down with a DISK FULL error! When the disk fills up, SPELL'N FIX automatically closes the output file and asks for another disk on which to continue!

In addition, you can perform most FLEX commands directly from SPELL'N FIX. The SSB version includes a directory list command which allows you to list disk files directly from SPELL'N FIX. This means that you don't have to stop and restart simply because you have forgotten a file name in the course of proofreading!

SPELL'N FIX is available for SSB DOS68 versions 5.1 and 6.0, and DOS69 (all versions), FLEX-p, 6809 FLEX (all versions except mini-FLEX), and can be ordered on 5- or 8-inch disk. Versions for 6809 FLEX and FHRCON available directly from STAR-KITS. OS-9 versions will be available soon.

ALFORD & ASSOCIATES
P.O. BOX 6683
RICHMOND, VA., 23230
804-320-6722

SSB UTILITIES...

We are now shipping our UTILITIES #1 disk for DOS68 and DOS69 users. If your disk library is getting out of hand, or if you are daily bombarded trying to make sense out of fifty, one hundred, or more files on a disk, this set of programs is for you!

We have developed a method to write a unique I.D. sector on SSB disks. In this sector, we include an 8-character disk access code, a 16-character disk name, an 8-character disk I.D., an 8-character disk creation date, an 8-character disk update date, and a 64-character comment field. Our BASIC utility prompts you for the information and builds the information sector.

CREATE then allows you to change the I.D. sector selectively. You can, without the access code, change only the update date and comment field, allowing you to make a quick change to reflect, say, the most recent changes to the files on the disk. Using the ACCESS code, you can change everything except the disk creation date.

LOCK allows you to list, delete, and write-protect your files. With list protection, the list-locked files only list when you supply the disk access code!

LIST is a multi-option program, allowing you to list tranlients, non-transients, locked files (if you supply the disk access code), free space, disk information, or any combination of these.

PURGE is a quick and easy mass file deletion program. Each file on the disk is presented for your inspection. Multiple presses then determine whether or not to delete each file.

TITLE generates BIG LETTER banner pages for your printed listings. You supply a name for the banner page of up to twelve characters, and TITLE generates the banner page with the current date and time (as contained in TOD#8 and TOD#9).

DUMP is a formatted, sideclued and ASCII memory dump. When directed to a hard copy device, page boundaries are honored to make things more readable.

COUNT allows you to quickly determine file sizes in lines and characters. Multiple files may be specified to allow you to count in a single pass those jobs which you have spread over several files.

UTILITIES #1 is available for DOS68, versions 5.1 and 6.0, and for DOS69 (all versions), and may be ordered on 5- or 8-inch disk. The manual also includes some very good information on the chained sector technique which SSB uses on their disks.

PRICES...

TREK-68/6800 disk, manual	$ 29.95
TREK-68/6809 disk, manual	$ 29.95
TREK-68 manual only	$ 12.95
COLOR-TREK cassette, manual	$ 17.95
COLOR-TREK- cassette, manual	$ 24.95
COLOR-TREK manual only	$ 5.95
SPELL'N FIX disk, manual	$ 69.29
UTILITIES #1 disk, manual	$ 29.95
UTILITIES #1 manual only	$ 12.95
VS-1 complete, disk, manual	$184.95
VS-1 complete, disk, manual (less SC-01A)	$154.95
VS-1 disk, manual (use VOX-EDIT with brand X)	$ 39.95
VS-1 manual only	$ 19.95
SC-01A sPEECHmaker chip	$ 45.00
SP-1 pack, cassette, manual	$179.95
SP-1 manual only	$ 14.95
SC-1/6800 disk, manual	$ 39.95
SC-1/6809 disk, manual	$ 39.95
SC-1/COLOR cassette, manual	$ 29.95
SCREDITOR III disk, manual, tutorial cassette	$150.00
SCREDITOR III manual and cassette only	$ 29.95

THE BEST NEWS...

God is truly a God of miracles. We could not begin to enumerate the miracles which He has performed for us, and wish to give thanks for His love and care in our lives. Without Him, we would not be in business. We hope that our honesty and devotion to our customers reflects our love for Him. We do make mistakes, and occasionally an order is loosed up or lost. But we'll do everything we can to correct any mistakes that we make. Thank you for your patronage.

SCREDITOR III DOES IT ALL...

This ad is being prepared using the SCREDITOR III word processor. Unlike any other editing package available for 6800/6809 machines, SCREDITOR III supports true multiple column editing capability. This ad was typed a column at a time. We cut and paste was used to prepare it (well, one, actually... our printer only prints 132 columns wide!).

SCREDITOR III is a true screen editor. What you are reading now is what we see as we were entering the text! With full cursor movement capability, automatic justification, true multi-column insert, delete, paragraph splitting and welding, and edit capability of up to 231 character positions per line, SCREDITOR III makes document preparation a breeze! Also, you can define page headers and headers, automatically, delete and then re-insert them, or change them at any time.

You can merge multiple files with SCREDITOR III. You can even specify where to insert a file, what line in the file to start from, and how many lines of the file to use! You can extract or copy lines from the current file into another file. You can even do form letter printing with automatic line item merging for custom mailings.

Automatic justification (left, center, right, or full line) can be turned on or off at any time. MARGINS can be moved at any time, just about any job can be done better with SCREDITOR III.

You can define "symbols" with SCREDITOR III. Up to 231 keystrokes may be included in a single symbol, including all edit and line commands, and later recalled with a single keystroke. Up to 26 symbols may be defined, and the symbols may be saved to a disk file for later recall.

Unlimited tabs may be set with SCREDITOR III, and may be changed at any time as you desire. Margins and tabs are continuously displayed for quick reference, along with all major edit modes, current column, current line, total lines, current display columns, etc...

SCREDITOR III runs with practically any SERIAL TERMINAL or with most any MEMORY MAPPED DISPLAY. If you change from a serial terminal to a memory-mapped display, or if you simply change display or terminal types, you don't have to buy a new copy of SCREDITOR III. Just run STYLER and define your new hardware! Even key definitions may be changed by STYLER! Terminals with special function keys may also be accommodated, simply run STYLER, and hit each key as it asks you to, and SCREDITOR III will use that key in your system!

SCREDITOR III is designed to provide capabilities for any office, personal, programming, or other editing and formatting job that you might have. The applications are essentially unlimited.

The manual provides extremely detailed information about every mode, feature and function of SCREDITOR III, and can entice with office personnel and the sophisticated user in mind. A unique feature of SCREDITOR III is a tutorial cassette which makes learning to use SCREDITOR III simpler than any other such program on the market. The cassette takes you step-by-step, through every function of SCREDITOR III, giving you practical examples and learning sessions with test files. No more reading for three weeks to learn how to do it!

SCREDITOR III is available to run under DOS68 versions 5.1 and 6.0 and all versions of 6800 FLEX (except mini-FLEX), and for DOS69, FLEX-p and OS-9. SCREDITOR III may be ordered on 5- or 8- inch disk.

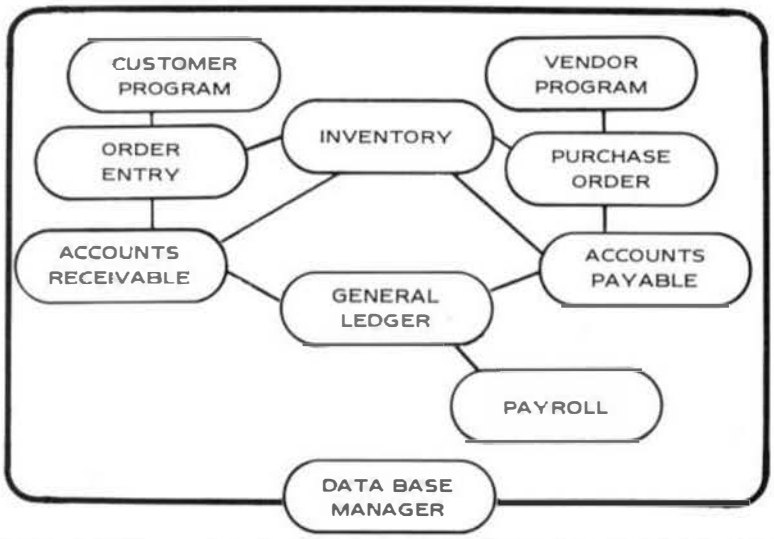
GENERAL INFORMATION...

Unless specified, all programs (except Color Computer) shipped on 5-inch disk for FLEX-p. Virginia orders add 4% sales tax. We ship by UPS on all domestic orders unless otherwise specified. COD's MUST go by UPS. Overseas orders shipped by air parcel post. On North American orders under $100, add $3 for shipping & handling. We pay shipping on orders over $100. On overseas orders, regardless of order value, add $10 for shipping. ORDERS RECEIVED WITHOUT SHIPPING WILL BE RETURNED UNFILLED.

We accept MASTERCHARGE, VISA as personal checks. Personal checks may take two to three weeks to clear before processing can proceed. All checks must be in U.S. funds drawn on U.S. banks. Dealer and OEM pricing and licensing agreements are available. Please CALL for details. Open accounts to Fortune 1000's, schools, established accounts only. We regret this policy, but can no longer finance other businesses).

FLEX / UniFLEX SOFTWARE

THE TOTAL SYSTEM



STAND ALONE SOFTWARE:

- VENDOR PROGRAM
- PURCHASE ORDER
- ACCTS. PAYABLE
- CUSTOMER PROGRAM
- ORDER ENTRY
- ACCTS. RECEIVABLE
- PAYROLL
- GENERAL LEDGER
- INVENTORY (Retail)
- INVENTORY (Manufacturer)
- M.F.I. (Extensive Mfg./P.O.)
- BULK MAILINGS
- MAINTENANCE SCHEDULE
- DATA BASE MGR.
- DATA BASE (Ulilities)
- SINGLE ENTRY LEDGER
- CHECK BALANCER
- COSTING & ESTIMATING
- CHURCH CONTRIBUTION PKG.
- CHURCH/SCHOOL BILLING
- FUND RAISING
- CUSTOM PROGRAMMING

INQUIRE ABOUT OUR COLOR COMPUTER SOFTWARE

SOFTWARE AVAILABLE THROUGH THESE DEALERS

Barney & Associates 113 W. Fifth Pittsburg, Kansas
Bonelli Media Assoc. 1116 N. 38th St. Allentown, Pa. 18104
Data Comp. P.O. Box 794 Hixson, Tn. 37343
Digital Services 1509 Tieton Dr. Yakima, Wa. 98902
RAM Associates 49 Overlook Dr. Framingham, Mass. 01701
MRLectronics B.V. Buitendofdreef 280 2625 Re delft. The Netherlands
New Technology Systems P.O. Box 2 Canatlan, DGO Mexico
Paris Radio 7 A Burton St. Darlinghurst, NSW Australia
Oowneast Data Box 416 Harborside Rd. Northeast Harbor, Me. 04662
SWTP 38 Dover St. London, England

2457 Wehrle Drive • Buffalo, New York 14221 • (716) 631-3011

**FLEX AND UNIFLEX ARE TRADEMARKS OF TECHNICAL SYSTEMS CONSULTANTS.*

UNIVERSAL DATA RESEARCH INC.

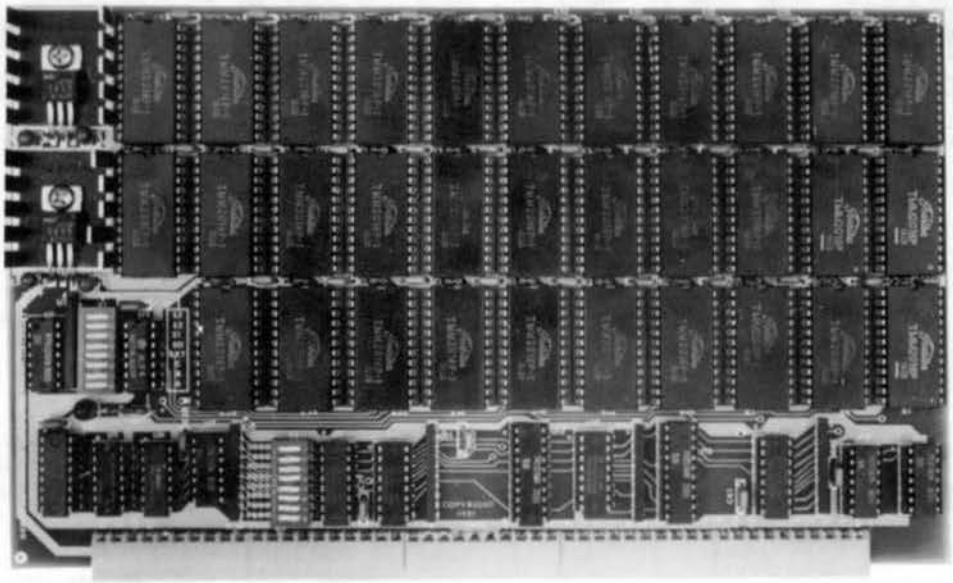
64K SS-50 STATIC RAM

NEW!

$259.00 (48K KIT)

NEW!

LOW POWER!



RAM OR EPROM!

BLANK PC BOARD WITH DOCUMENTATION $52

SUPPORT ICs + CAPS -	$18.00
FULL SOCKET SET -	$15.00

ASSEMBLED AND TESTED **ADD $40**

FEATURES:

- ★ Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- ★ Fully supports Extended Addressing.
- ★ 64K draws only approximately 500 MA.
- ★ 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- ★ Board is configured as 3-16K blocks and 8-2K blocks (within any 64K block) for maximum flexibility.
- ★ 2716 EPROMs may be installed anywhere on Board.
- ★ Top 16K may be disabled in 2K blocks to avoid any I/O conflicts.
- ★ One Board supports both RAM and EPROM.
- ★ RAM supports 2MHZ operation at no extra charge!
- ★ Board may be partially populated in 16K increments.

56K Kit	$309
64K Kit	$359

16K STATIC RAMS?

The new 2K x 8, 24 PIN, static RAMs are the next generation of high density, high speed, low power, RAMs. Pioneered by such companies as HITACHI and TOSHIBA, and soon to be second sourced by most major U.S. manufacturers, these ultra low power parts, feature 2716 compatible pin out. Thus fully interchangeable ROM/RAM boards are at last a reality, and you get BLINDING speed and LOW power thrown in for virtually nothing.

Digital Research Computers
(OF TEXAS)
P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Add $2.00 postage. We pay balance. Order under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50, add 85¢ for insurance.

FBASIC FOR 6800 AND 6809

Finally an alternative to assembly language for fast smoothly animated graphics. FBASIC is an ultra fast BASIC compiler with "Verbs" and "Functions" to support all ARCADE 50 features. This is a true compiler that produces optimized machine language modules which are ROMable and require no Run-Time package. FBASIC requires less memory overhead and runs hundreds of times faster than BASIC interpreters. It supports standard BASIC instruction including String functions, Disk I/O and fast integer arithmetic with multiple-precision capability. Many new commands have been added to support the ARCADE 50. Graphic commands include MOVE, RMOVE, DRAW, RDRAW, VPRINT, VPOKE, etc. Sound commands are TONE, NOTE and ENVEL. ADC, JSWITCH, SWITCH and VPEEK are some of the new functions.

ARCADE 50, assembled and tested	$325.00
Video and Audio connector set	15.00
4 Joystick connector set (5 pin DIN)	15.00
2 Radio Shack Joysticks	24.00
UHF channel 33 modulator	32.00
Gold Molex connectors	12.00
FBASIC (specify 6800 or 6809)	$110.00
FBASIC (with ARCADE 50)	75.00
ARCADE 80 (TRS 80 MODEL I)	$395.00
LABVIDEO (Motorola EXORbus)	375.00
6809 Processor Board	225.00

- Comes assembled with PIA and ACIA
- 12 Sockets for 2716, 2732 or RAM
- Supports DAM disk I/O
- No extended memory or DAT
- Ideal for 6809 upgrade or process control

TERMINUS DESIGN INC
16 SCARBROUGH ROAD
ELLENWOOD, GA 30049
(404) 474-4866

PRODUCTS IN THE MIX:
RGB ARCADE 50 ADAPTOR
ARCADE 100 (S-100)
DMA Disk Controller

ARCADE 50

POWERFUL COLOR GRAPHICS

Uses the new TMS9918A Video Display Processor
High resolution 256 x 192 pixel display with 15 colors
16k Bytes of onboard RAM does not reduce user memory
32 graphic images can be individually moved with simple X-Y commands for smooth animation
External video input allows subtitling
NTSC composite video output

SOUND EFFECTS AND MUSIC

Three AY3-8910 Programmable Sound Generators
Nine simultaneous voices
Three independent noise sources
Onboard stereo amplifier drives two 8 ohmn speakers

ADDITIONAL I/O CAPABILITIES

Eight analog inputs with 8 bit resolution
Supports four joysticks with pushbutton switches
Eight bit parallel I/O port
Entire unit maps into 256 bytes of memory

DOCUMENTATION AND SOFTWARE

Programming manuals for Video and Sound Processors
Subroutine library and Super Demo Maze Game
Example programs in BASIC, FBASIC and ASSEMBLY
User library and sales support

Specify 5" or 8" soft sector disk for TSC's FLEX or MICROWARE'S OS/9 system.
TERMS: CASH, VISA, MC., C.O.D.

5 MEG HARD DISK $600.00

This is a new Shugart SA-10002 5.33 megabyte hard disk drive. Interfacing is a snap with the Western Digital WD1000-80 intelligent controller card ($500). An example of the interface and software can be found in next month's issue. Power supply ($150) is good for 2 drives and controller. Interfaces for CP/M, S-100, and STD BUS are also available.

Computer DYNAMICS
105 S. Main St. Greer, SC 29651
Ph. 803-877-7471

6809 C Compiler

VERSION 2.0

- Generates assembly language source output
- Requires only 20K memory
- Extensive library functions in source code
- Generates position independent code
- Supports most C functions
- Generates ROMmable code
- Floating point

Specify diskette size SSB, FLEX, or OS-9

VERSION 2.0 Price $120.00
The C Programming Language—Kernighan & Ritchie
Price $16.00

TERMS: Check, MC/Visa. California residents add 6%. Foreign or C.O.D. add 15% for shipping and handling.

(714) 755-4373
DUGGER'S GROWING SYSTEMS
POST OFFICE BOX 305 SOLANA BEACH, CA 92075

ELEKTRA COMPUTER PRODUCTS

ELEKTRA CABINET

Made of heavyweight 0.090" thick aluminum.
Interior Size: 18-1/2" wide by 21-7/8" deep x 6-3/4" high
Heavy duty A.C. line cord.
A.C. fuse holder
EMI filter
Fan with filter
Back panel has 10 cutouts for "D" type data connectors
Front panel has key on/off power switch, 2 illuminated push buttons (Reset and NMI/Abort), and 2 cutouts for 5-1/4" floppy drives

Without power supply:	$260.00
With 110v power supply, disk regulator board or two filler plates:	$460.00
With 220v power supply, disk regulator board or two filler plates:	$510.00

POWER SUPPLY

Highest quality linear power supply conservatively rated at 15a @ 8v, 3a @ 16v, 3a @ -16v
3 primary inputs to adjust for light, medium, and heavy loading

110v version: $175.00 220v version: $225.00

DISK REGULATOR BOARD with cables

Powers 2 5-1/4" floppy drives. $50

Filter (fan): $10.00

ELEKTRA CPU 8/9

Choice of 6802, 6808 or 6809 CPU
(6802 and 6808 are software compatible with the 6800 at the opcode level).

DEVICE		6809 ADDRESS	6802-6808 ADDRESS
3 2716 Eproms	Eprom #3	F800-FFFF	F800-FFFF and E000-E7FF
	Eprom #2	F000-F7FF	F000-F7FF
	Eprom #1	E800-EFFF	E800-EFFF
1K Scratchpad RAM		E400-E7FF	A400-A7FF and A000-A3FF
MC6840 Triple Timer		E210-E217	8200-8207

MC14411 Baud Rate Generator producing baud rates of:
Low Range 110, 150, 300, 600, 1200, 4800, and 9600
High Range 440, 600, 1200, 2400, 4800, 19200, and 38400
The board does not contain a DAT and does not support extended addressing.
The board supports DMA by either HALT or BUSREQ when a 6809 CPU is used.
DMA to the devices on the CPU card is not supported.
The board will run any of the MIKBUG™ compatible monitors in the 6802-6808 mode and SBUG-E, NUMBUG, and GIMIXBUG-09 in the 6809 mode. The ELECTRA CPU 8/9 will run any of the popular disk controller boards with the appropriate software. Special versions of OS-9™ L1 are available.

Bare board: $50.00* **Kit: $225.00*** **Assembled: $275.00**

ELECTRA DPS Dual Port Serial Card

Fits the standard 30 pin SS-50 bus I/O slot
Can be configured for 4 addresses per port with the B port 2 addresses higher than the A port or for 16 addresses per port with the B port 4 addresses higher than the A port.
Each port is terminated at two 16 pin dip sockets, one socket configured for modem and the other socket configured for terminal or printer. RTS, CTS, DTR, DCD, OTR are appropriately implemented.
Each port has independent selection of baud rate.
Each port allows the interrupt request to be jumpered to the IRQ or FIRQ/NMI bus line.

Bare board: $20.00* **Kit: $60.00*** **Assembled: $80.00**

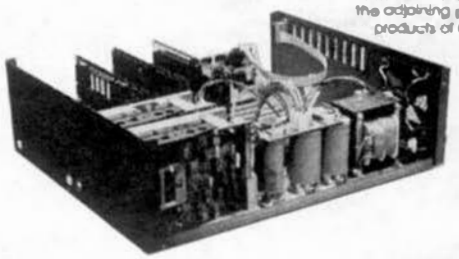
Assembled cable (two required for each interface board) $20.00 each

ELEKTRA DPP Dual Port Parallel Card

Fits the standard 30 pin SS-50 bus I/O slot.
Can be configured for 4 addresses per port or 16 addresses per port (occupying the first four addresses of the I/O slot)
The direction of the TTL buffers can be controlled by either on board jumper selection or by a signal from the peripherals.
The interrupt request lines for each port may be individually jumpered to the IRQ or FIRQ/NMI bus line.

Bare board: $20.00* **Kit: $60.00*** **Assembled: $80.00**

Assembled cable (two required for each interface board) $20.00 each



The CPU, 56K memory board, and DMA controller board in the adjoining picture are products of GIMIX, Inc.

ELEKTRA Motherboard

Heavyweight 0.125" thick. 18" long by 9" wide.
11 memory (50 pin) slots. 4 or 8 slots may be cut off for shortening to 14" and 10" lengths respectively.
8 I/O (30) pin slots.
1" spacing between all memory and I/O slots.
On board baud rate generator with high and low ranges providing jumper selectable rates for each of the five I/O baud rate lines.
Complete address decoding and selectability for the I/O ports.
Choice of 4, 8, or 16 addresses per I/O port.
Slow device circuitry permitting 1 MHz 30 pin disk controllers to run with 2 MHz 50 pin CPU boards.
Extended addressing capability for meeting SS-50C bus specifications.

Bare board: $80.00 **Kit: $240.00**** **Assembled: $300.00****

"Gold square pin connectors instead of tin add $80.00

ELEKTRA Chassis

Includes cabinet, power supply, disk regulator board and power cable, and assembled motherboard with gold connectors, totally installed. Ready to use with documentation. 5025 Add $50.00 for 220V.

GIMIX 16K memory board with control registers	$195.00
GIMIX #15 memory board with 32K static RAM	$295.00

*WARNING

AAA Chicago Computer Center does not provide repair or diagnostic service for customer assembled kits. AAA Chicago Computer Center does warranty and maintain service for our assembled boards. The customer should carefully take into consideration the small differential separating our kit and assembled prices when making his choice of purchase.

We have introduced our line of computer equipment with the purpose of offering the highest quality of components possible at affordable prices. These products are intended for OEM applications where it is the responsibility of the purchaser to integrate these components with suitable memory, disk controller(s), drives, and software along with I/O terminal(s) to form working computer system(s).

AAA Chicago Computer Center

120 Chestnut Lane • Wheeling, IL 60090 • (312) 459-0450

Phone consultation available most weekdays from 4 PM to 6 PM

SMOOTH™ Software

SUPER MODEM PROGRAM

Transmit manu lly to distant c puter

Transmit disk files (text) of any length to distant computer

Receive and save disk files (text) of any length on local disk system. If sending computer does not support an X-on/X-off protocol, then the received files are limited in size by the computer memory

Tested to transmit and receive text at speeds up to 9600 baud. (CRT term nal must be capable of operating at a baud rate higher than the one the modem is operated at.)

Half duplex option in case distant computer doesn't echo

Echo option so user can simulate a time sharing system. (Super Modem Program doesn't support auto-answer but the source is provided for those individuals who wish to adapt our program to their special needs.)

Replaces CR with CR/LF (user option) for those using time sharing systems that don't transmit LF's

Slow disk file transmit (user option) based on character verification for use on time sharing systems to which disk files cannot be sent at speed suggested by the baud rate

Please specify 6800 SSB, 6800 FLEX™, or 6809 FLEX™, 5" or 8"
Manual and disk with both source and object code $75.00

STANDARD MODEM PROGRAM

Same as Super Modem Program above but without ECHO option, CR/LF for CR option, slow disk file transmit option nor X-on/X-off option. Reception of disk files is limited to those small enough to compile or fit within the receiving buffer.

Please specify 6800 SSB, 6800 FLEX™, or 6809 FLEX™, 5" or 8"
Manual with instructions, source listing, and flow chart, disk with both source and object code $45.00

ALL IN ONE
Editor - Text Processor - Mailing Labels
Mailing Lists - Use any CRT terminal and printer

Supports Editing commands such as bottom, change, delete, find, insert (single line), input (multiple lines), list, read, overlay (with cursor editing, character deletion and insertion), overstrike (for selected parts of text), print, restart, set, top, underline, up and verify

Supports Text Processing commands such as block copy, block move, centering, margin justification (widen and narrow), paging, and tabbing

Mailing Lists and Labels. Use the same mailing list disk file (with protected areas) for both mailing labels and repeat letters. Repeat letters are personally adressed to each person or selected persons on the mailing list.

Most Powerful File Handler found in any editor. Append one file to the end of another, or insert (merge) one file into another as designated by the line pointer. Print specified lines to your printer or to a disk file. Edit files larger than the text buffer. Does not produce output files when not desired. Delete disk files from the editor.

Printer commands. Control characters can be sent to the printer for format control either directly from the control terminal or by imbedding them in the text. The set command contains interface initialization and character output routines to support the SWTPC MP-C interface as well as the standard serial and parallel interfaces. Jumps are also provided to user supplied printer routines. User selects the port address (0 thru 7, A or B) thereby eliminating the need for the user to install printer software routines. Editor can be initialized for either 4 or 16 addresses per port.

Editor allows exiting to either the monitor or DOS and then reenter (Warm Start) without destroying previously prepared text in the buffer. The Restart command erases contents in the buffer without the user having to reload the Editor.

The Editor allows the user to toggle between full duplex (no echo) and half duplex (echo) as needed. It responds to commands in both upper and lower case and can be used to create assembler source code and Basic programs as well as text.

Specify 6800 SSB, 6800 FLEX™ 6809 FLEX™, 5" or 8" 50.00
Printed source listing is available for an additional 35.00

Software by Technical Systems Consultants, Inc.

Flex™ (includes Editor and Assembler)	150.00
UniFLEX™ (includes one year maintenance and update)	450.00
Editor	50.00
Assembler	50.00
68000 Cross Assembler on 6809	250.00
6809 Cross Assembler on 6800	100.00
Text Processor	75.00
Extended Basic	100.00
Basic Precompiler (specify standard or extended)	50.00
Pascal (Flex™)	200.00
Pascal (UniFLEX™) (Add $75.00 for one year's maintenance and update)	225.00
Sort-Merge Package	75.00
6809 Flex™ Utilities	75.00
6800 Flex™ Utilities	100.00
Debug Package	75.00
Diagnostic Package	75.00

Software by Microware Systems Corp.	UPDATE	SOURCE	MANUAL	OBJECT
OS-9™ Level One Operating System	75.00	400.00	40.00	200.00
OS-9™ Level Two Operating System	75.00	N/A	40.00	500.00
BASIC09™	75.00	N/A	75.00	200.00
OS-9™ Macro Text Editor		300.00	15.00	125.00
OS-9™ Interactive Assembler		300.00	10.00	125.00
OS-9™ Interactive Debugger (Disk version)		100.00	10.00	50.00
CIS Cobol Compiler	250.00	N/A	60.00	600.00
Pascal Compiler	100.00	N/A	40.00	400.00

SWTPC

	Kit / Assembled
DMF2 Disk Controller Board (NEW)	95.00
DMF2 Disk Controller Board (Used, very limited quantity)	450.00
SWTPC FLEX™ Disk and manual	35.00
SWTPC FLEX™ Disk without manual	10.00
DC-2 Disk Controller (Limited Quantity)	N/A / 125.00
4K RAM Board (Limited Quantity)	N/A / 40.00
SBUG-E (2716 compatible)	19.95
SWT8UG (6830 compatible, limited Quantity)	N/A
MP-A2 6800 CPU Board	150.00
MP-S2 Serial interface (dual port)	N/A / 120.00
MP-LA Parallel interface (dual port, limited quantity)	40.00 / 80.00
MP-L2 Parallel interface (dual port)	N/A / 120.00
MP-R Single voltage 2716 prom programmer	N/A / 114.50
MP-N Calculator board	54.95 / 92.00
MP-T Interrupt timer	N/A / 92.00
MP-8M 8K 4044 Memory board (limited quantity of kits)	150.00 / 275.00
S32 Universal Static Memory Board	N/A / 124.50
MP-09 6809 CPU board	N/A / 295.00
69 Chassis, P.S., 68B09 CPU, 8K, RAM, One Serial Port	N/A / 799.00

Universal 68XX Bare Motherboard. 6800/6809, 4/16 addresses per port, 8 50 pin/8 30 pin slots, baud rate generator, 15 1/8" x 9 3/8" 65.00
ELEKTRA Motherboard (bare) 80.00
Connectors (10 pin, Titanium-Tin plated 5 microns for near gold quality)
Male with square cross section pins ..... each .50
Female ..... each .75
Gold, Male w/square pins or female ..... each 1.50

SMOOTH™ and **ELEKTRA™** are trademarks of AAA Chicago Computer Center
FLEX and **UniFLEX** are trademarks of Technical Systems Consultants, Inc.
OS-9 and **BASIC09** are trademarks of Microware Systems Corp.
GIMIX® and **GHOST®** are registered trademarks of GIMIX Inc.

ELEKTRA® S550 Computer Products

ELEKTRA Dual drive cabinet for 5-1/4" drives with power supply, line cord, fuse, power switch, and power cable to drives	125.00
ELEKTRA Dual drive cabinet, power supply, ps cable for 8" drives	350.00
Cabinet for dual 8" drives only	250.00
Power supply for dual 8" drives only	120.00
PS cables only (Specify brand and type of 8" drives)	30.00

Special Software

4K 6809 HUMBUG	75.00
4K 6800 HUMBUG (RAM needed at $A000 and $D000)	65.00
2K 6800 HUMBUG (With cassette LOAD and PUNCH)	40.00
2K 6800 HUMBUG (Extra commands instead of cassette software)	40.00
Other HUMBUG versions including video versions are available	
Spell'n Fix by Peter Stark	89.29
Dynamite Disassembler	60.00
SUPER SLEUTH Disassembler System	89.00

DISK DRIVES 30 day guarantee, single and double density capability

5-1/4", single head, 40 tracks	$250.00
5-1/4", single head, 40 tracks, flippy	260.00
5-1/4", double head, 40 x 2 tracks	335.00
5-1/4", single head, 80 tracks	335.00
5-1/4", single head, 80 tr cks, flippy	345.00
5-1/4", double head, 80 X 2 tracks	470.00
MPI - Service Manual	20.00
Siemens Manual	10.00
8", single head, 77 tracks	375.00
8", single head, 77 tracks, Thin-Line	450.00
8", double head, 77 X 2 tracks	495.00
8", double head, 77 X 2 tracks, Thin-Line	25.00
Microtime 6800 Calendar and Clock Board (assembled and tested)	105.00
Bareboard, connector, and documentation only of above (See review Feb. 1980 68 Micro Journal)	35.00
Microtime II	75.00
Data Mart 16K EPROM barebo rd (2708 chips)	30.00

Printers

Epson MX-80 (Centronics compatible, parallel interface)	439.00
(with Serial RS-232 interface option)	add 75.00
Spare Print Head	39.95
Spare ribbon cartridge	15.00
C. Itoh Comet I 1 5 cps, 9 x 7, bidirectional, serial or parallel	295.00
Okidata Microline 82A 120 cps, 9 x 9, bidirectional, serial and parallel	495.00
Tractor for 82A	70.00
Optimal Technology, Inc. EP-2A-79 Eprom Programmer	169.00
(Personality Modules extra for above programmer)	
Optimal Technology, Inc. 30 pin parallel I/O Board for EP-2A-79	35.00
Software package for EP-2A-79 (Specify 6800 or 6809)	29.95
EP-2A-78-M-01 Eprom Programmer (User configures and supplies power)	79.95

Smoke Signal Broadcasting

Monitor on 2716 with manual (Specify Chelsea or SWTPC)	75.00
DCB-4A Double Density Controller Board for 5" and 8" with DOS	549.00
DOS69D DOS Update with Editor and Assembler (Specify 5" or 8")	75.00
LMB-1A Motherboard	399.00
SCB-69 6809 CPU Board	399.00
M-16-X 16K Static Memory Board	195.00
M-24-X 24K Static Memory Board	295.00
M-32-X 32K Static Memory Board	395.00

GIMIX

2 MHz 6809 Plus CPU, time of day clock, battery backup, 1K NMOS RAM	578.05
CMOS RAM substitution	8.00
GIMIX Dynamic Address Translator	35.00
SWPTC compatible DAT	15.00
Missing cycle detect card	38.23
Disk Controllers (All have data separators and can be used with either single or double headed drives)	
5" single density controller without 1771 chip	158.38
5" single density controller complete	198.48
5" and 8" single density controller complete	226.58
5" double density controller with variable precomp	298.28
GMA 5" AND 8" double density controller with variable precomp	588.68
GIMIX version of FLEX™ (without Editor and Assembler)	90.00
Double disk regulator card	68.22
Ribbon cable for two 5 1/4" disk drives (short)	34.96
Ribbon cable for two 5 1/4" disk drives (long)	44.96
Ribbon cable for two 8" disk drives (long)	44.26
8" disk drive cabinet with power suppl	848.18

Memory	CMOS WITH BAT BACKUP	NMOS NO BAT BACKUP
16K Static RAM Board with control registers**		195.00
32K Static RAM Board with 32K of RAM installed*		295.00
**discontinued, limited quantity available		
64K Static RAM Board with 24K of RAM installed	N/A	348.27
64K Static RAM Board with 32K of RAM installed	518.36	398.37
64K Static RAM Board with 48K of RAM installed	N/A	518.47
64K Static RAM Board with 56K of RAM installed	728.56	578.57
64K Static RAM Board with 64K of RAM installed	798.64	638.67
16 Socket EPROM/ROM/RAM Board		238.32
8K Promboard (2708)		98.34
4K PPD 4K Prom Board and 2708 Prom Burner		198.00

I/O Boards

Single port 30 pin serial interface (Requires 1 cable set)	88.41
Dual port 30 pin serial interface (Requires 2 cable sets)	128.43
8 port 50 pin serial interface with baud rate generator	318.46
Dual port 30 pin parallel interface (Requires 2 cable sets)	88.42
8 port 50 pin parallel interface with interrupt generator	198.45
Cable sets for above boards (specify board)	24.95
2MHz 6809 PLUS Computer System with 56K Memory*	2498.29
Above System with #58 Controller and Special Software Pkg.*	2988.59
Above System with #68 Controller and Special Software Pkg.*	3248.49
*with CMOS RAM and Battery Backup	add 300.00
Mainframe (Chassis, PS, Switches, Fan, Motherboard, Baud Rate Gen.)	1198.19

Shipping and handling estimates

Within the Continental U.S., please add 3% ($5.00 minimum)
Foreign, prepaid and add 10% ($10.00 minimum) for light items only
Heavy items must be prepaid and will be shipped Emery Air Freight Collect

Please phone during consultation hours if questions arise regarding shipping fees

Master Charge, Visa, and American Express honored

AAA Chicago Computer Center

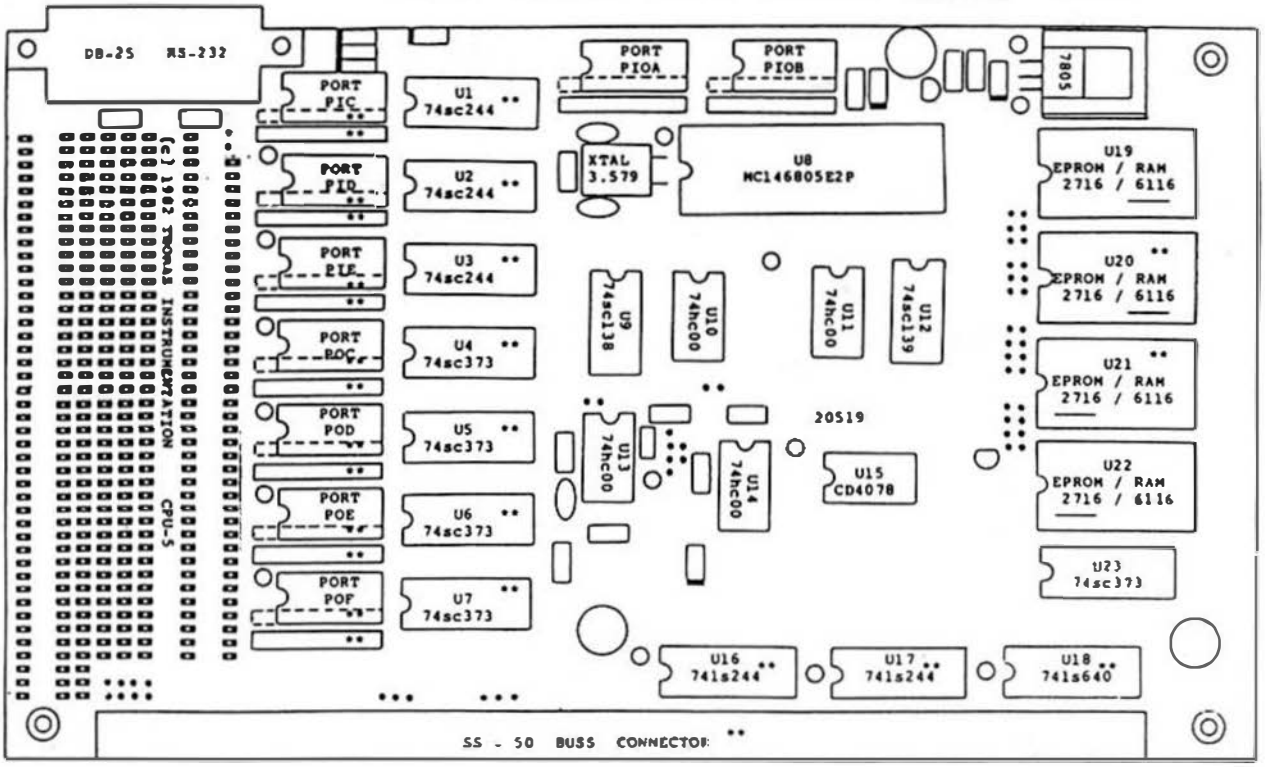
120 Chestnut Lane, Wheeling, IL 60090

(312) 459-0450 Phone Consultation available most weekdays from 4 PM to 6 PM

Our apology. We are running 4 weeks behind answering our mail regarding technical questions. Please phone during consultation hours for technical help.

Prices and inventory are subject to change without advance notice
This ad is our catalog

*** NEW 6805 ALL CMOS CARD ***



CPU-5-B	Blank card with documentation	$49.00
CPU-5-P	Partial card (parts marked ** not incl.)	$229.00 } Includes EPROM Monitor
CPU-5-F	Full card with all parts shown (fully assembled)	$349.00 } and 115 vac Power Pak.
CROSS-ASSEMBLERS	for FLEX (c) TSC 6809 system	$150.00

SOFTWARE

VDISK Treat extended memory as a super fast disk drive

6809 source & object	$149.00
6809 object	$ 99.00

OUTSIDE MODEM PROGRAM incl. source

UniFLEX version	$100.00
FLEX version 6800 & 6809	$ 50.00
HAYES SMART MODEM	$249.00

CROSS ASSEMBLERS for 6800,6801,6805 runs on 6809 FLEX $150.00

TV-EDIT Screen oriented Editor for FLEX 6809 $ 95.00
Please specify 5" or 8" Disk when placing order.

CHIPS

Please note when ordering IC's that a $100.00 Minimum is in effect Orders less than the minimum will be charged an additional $10.00 for handling

6116P-3 2MHZ	$13.50
2016P-2 2MHZ	$10.50
2716 1MHZ	$ 4.50
2716-1 2MHZ	$16.50
146805E2P	$25.00
68B02	$10.00
68B09	$25.00
68B21	$ 6.50
68B40	$12.00
68B50	$ 6.75
74LS640	$ 3.35

HARDWARE

NEW MEMORY PRICES TO INTRODUCE VDISK

S-R/R without memory chips	$120.00
with 8K NMOS@2MHZ	$169.00
with 16K NMOS@2MHZ	$199.00
with 32K NMOS@2MHZ	$299.00
with 48K NMOS@2MHZ	$399.00
Bare Card	$ 49.00

Extender Cards assemb. with logic aid

SS-50/50C	$ 35.00
SS-30	$ 25.00
SP-1 Prototype Board A/T	$195.00
SP-1 Bare Card	$ 49.00
SS-50 Wire Wrap Board B/C	$ 39.00
SS-30 Wire Wrap Board B/C	$ 20.00
6802 Super CPU A/T	$235.00
6802 Super CPU B/C	$ 59.00
Video RAM, B/C $49.00, A/T	$195.00
Parallel I/O, B/C $49.00, A/T	$139.00
SS-50 Backplanes 4,6,8,12,616 position @ $5.00 per slot w/o connectors	
SS-30 8 pos. BP w/o connect.	$ 39.00
Transition Card $49.00, A/T	$ 95.00
Molex Gold Male $1.60, Female	$ 1.60
Molex Tin Male $ .40, Female	$ .50

THOMAS INSTRUMENTATION

168 EIGHTH STREET — AVALON, N.J. 08202 (609) 967-4280
NJ RES. INCLUDE 5% SALES TAX
Allow 3 Weeks for Checks to Clear
CONT. USA INCLUDE $3.00 SHIPPING, CANADA $6.00, FOREIGN $12.00
MASTERCARD, VISA, and C.O.D. ACCEPTED

Send $20.00 to receive full documentation, schematics, & source listings for all boards currently in production.

'68' MICRO JOURNAL ADVERTISERS INDEX

This index is provided as a reader service. The publisher does not assume any liability for omissions or errors.

2MHZ 6809 SYSTEMS

GIMIX offers you a variety to choose from!

38 MB WINCHESTER SYSTEM $17,498.99

HARDWARE FEATURES:

- ★ 2MHz 6809 CPU
- ★ 512KB Static RAM
- ★ 8 RS232C Serial Ports
- ★ 2 Parallel Ports
- ★ DMA Double Density Floppy Disk Controller
- ★ Dual 8" DSDD Floppy Disk System
- ★ Dual Winchester Subsystem with Two19 MB 5¼" Winchester Drives

SOFTWARE FEATURES:

- ★ OS-9 LEVEL TWO Multi-User Operating System
- ★ OS-9 Debugger
- ★ OS-9 Text Editor
- ★ OS-9 Assembler

19 MB WINCHESTER SYSTEM $8998.09

HARDWARE FEATURES:

- ★ 128K Static Ram
- ★ 2MHz 6809 CPU
- ★ 19 MB 5¼" Winchester DMA Subsystem
- ★ 4 RS232C Serial Ports
- ★ 1 MB 5¼" Floppy Disk Drive
- ★ DMA Double Density Floppy Disk Controller

SOFTWARE FEATURES:

- ★ OS-9 LEVEL TWO Multi-User Operating System
- ★ OS-9 Text Editor
- ★ OS-9 Debugger
- ★ OS-9 Assembler

128KB MULTI-USER SYSTEM $6997.39

HARDWARE FEATURES:

- ★ 2MHz 6809 CPU
- ★ DMA Double Density Floppy Disk Controller
- ★ 128KB Static Ram
- ★ 2 RS232C Serial Ports
- ★ Dual 8" DSDD Floppy Disk System

SOFTWARE FEATURES: Your choice of either UniFLEX or OS-9 LEVEL TWO. Both are Unix-like Multi-User/Multi-Tasking Operating Systems.

56KB FLEX / OS-9 "SWITCHING" SYSTEM $4148.49

HARDWARE FEATURES:

- ★ 2MHz 6809 CPU
- ★ 56K Static Ram
- ★ 2 RS232C Serial Ports
- ★ DMA Double Density Floppy Disk Controller
- ★ 2 Built-in 5¼" 40tr DSDD Disk Drives (80 Track DSDD Drive Option . . add $400.00)

SOFTWARE FEATURES:

- ★ GMXBUG monitor — FLEX Disk Operating System
- ★ OS-9 LEVEL ONE Multi-tasking operating system for up to 56K of memory

WINCHESTER SUBSYSTEMS

Winchester packages are available for upgrading current **GIMIX** 6809 systems equipped with DMA controllers, at least one floppy disk drive, and running FLEX, OS-9 LEVEL ONE or OS-9 LEVEL TWO. The packages include one or two 19MB (unformatted) Winchester drives, DMA Hard Disk Interface, and the appropriate software drivers. The Interface can handle two 5¼" Winchester Drives, providing Automatic Data Error Detection and Correction: up to 22 bit burst error detection and 11 bit burst error correction.

Dual drives can be used together to provide over 30 MBytes of on line storage -- or use one for back-up of the other. (More convenient and reliable than tape backup systems.

#90 includes one 19MB Drive, Interface, and Software **$4288.90**
#91 includes two 19MB Drives, Interface, and Software. **$6688.91**

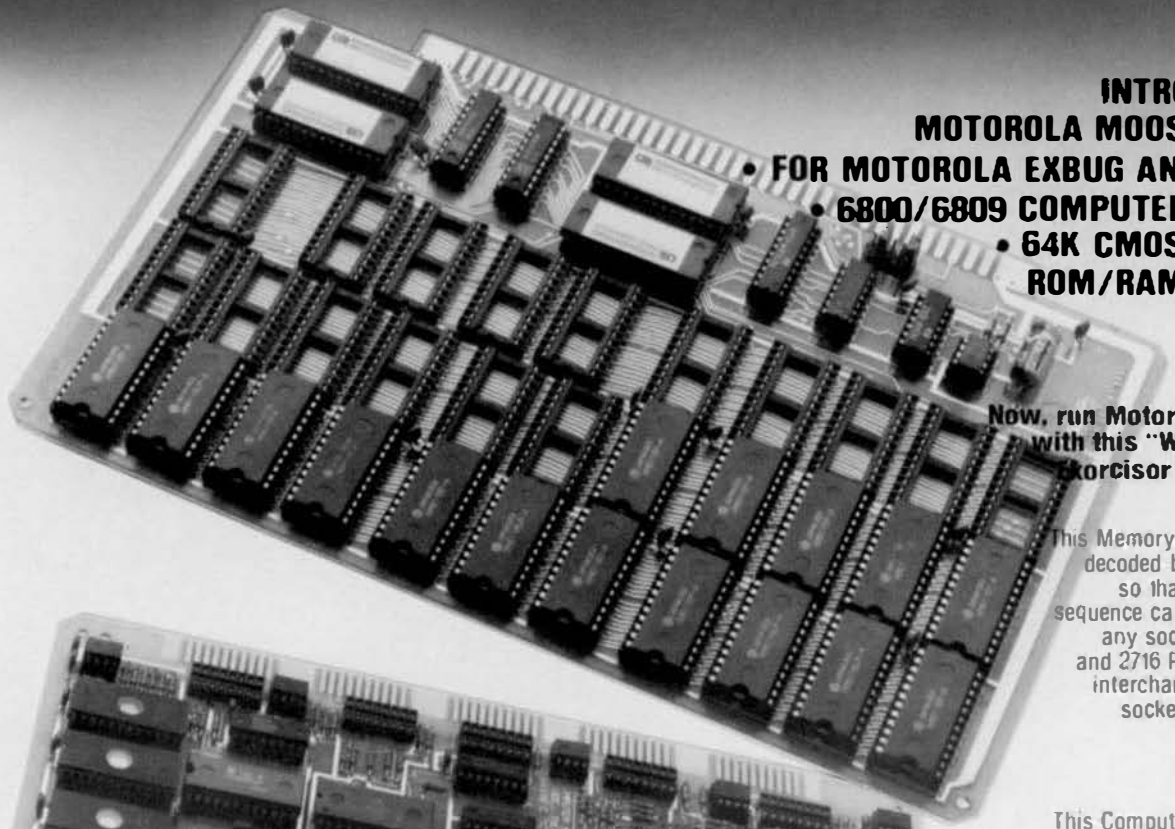
Contact GIMIX for systems customized to your needs or for more information.

50 HZ Export Versions Available

1337 WEST 37th PLACE
CHICAGO, ILLINOIS 60609
(312) 927-5510
TWX 910-221-4055

GIMIX INC.

THE WILD PAIR.



INTRODUCING
MOTOROLA MOOS BY CSA
• FOR MOTOROLA EXBUG AND MOOS!
• 6800/6809 COMPUTER BOARD!
• 64K CMOS, STATIC ROM/RAM BOARD!

Now, run Motorola's MDOS with this "Wild Pair" of Exorcisor Compatible Boards!

This Memory Board is fully decoded by four FPLA's so that any address sequence can be alloted to any socket. 6116 RAM and 2716 PROM are truly interchangeable in any socket. No jumpers.

Just $690.

This Computer Card sports five ACIA's for full handshake RS232 ports and sufficient ROM and RAM to support Motorola's Exbug and MDOS. All addressable devices are fully decoded by FPLA.

Only $595.

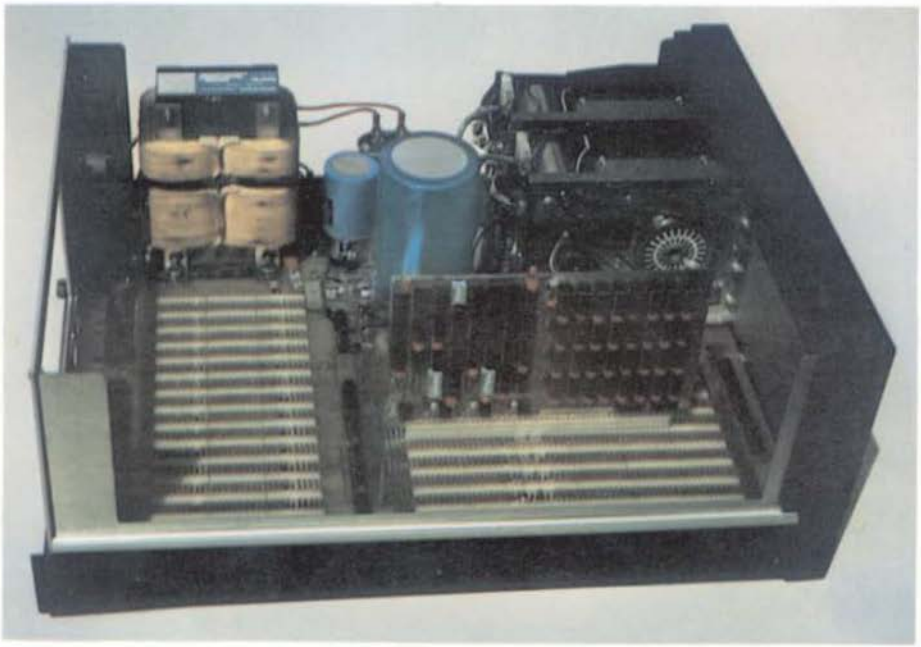
CSA

Computer System Associates, Inc.
7562 Trade Street, San Diego, CA. 92121; Phone (714) 566-3911.
Ask for Alfonso Bruno.

68 Micro Journal
5900 Cassandra Smith Rd.
Hixson, TN 37343

Second Class Postage Paid
At Chattanooga, TN
ISSN 0194-5025

HELIX™



THE MAINFRAME

- Industry Standard Optima™ Cabinet
- Largest Constant Voltage Power Supply in the industry
- S-64 Bus gives 16 Bit Power and S-50 Bus Compatibility
- 10 Main (S-64) Slots
- 14 I/O (S-30) Slots plus 2 On-board
- On-board Baud Rate Generator to 38.4Kb
- Space and Power for two 5¼" Disk Drives
- Full Address Decoding for I/O Slots
- Two RS-232 Serial and Two parallel Ports On-board
- Single Board Construction for Reliability
- Faraday Shielded Bus Lines give "Text Book Clean" Signals

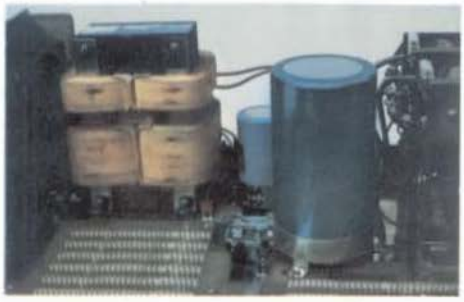
THE PROCESSORS

6809

- Standard 2 MHz Operation
- Standard DAT Compatible with GIMIX and SWTPC
- Standard 6840 Interval Timer
- Standard 1K Scratchpad RAM
- Standard Clock/Calendar with Battery
- Provision for Programmers Console

68000

- Standard 8 MHz Operation
- Memory Management Hardware
- Provision for Programmers Console
- 16 Bit Power and 8 Bit Compatibility



The HELIX™ computer system represents the latest advance in S-50 bus computer systems. Relying on the physical nature of S-50 bus connectors to guarantee compatibility, the HELIX adds 14 bus lines (becoming S-64) to allow a 68000 processor to operate with full 16 bit data transfer and 24 bit addressing, while at the same time providing full interchangability with existing S-50 components.

Offered with a selection of processors, memories, and peripheral controllers, a HELIX system can be configured for applications ranging from advanced hobbyist to multiterminal time-sharing.

Designed to offer the utmost in speed, reliability, and utility at a reasonable price, it represents a new standard of quality for those who require a professionally designed computer for professional use.

THE POWER SUPPLY

- Ferro-resonant Transformer for Line Noise and Under-Voltage Protection
- Conservative 25 Amps at 8.5 Volts
- Conservative 5 Amps at ±16 Volts
- Conservative Component Rating for Reliability

THE COMPONENTS

- Fully Socketed
- Gold Plated Bus Connectors
- Only "B" Series 68XX Components Used
- Only Top Grade Logic Circuits Used
- Industrial Grade Components Throughout

THE MEMORIES

DM-64

- Field Proven
- Proprietary Memory Control Logic
- Fully Transparent Refresh
- Tested at 2.5 MHz Operation

DM-512

- 512K Bytes on a Single S-64 Board
- 16 Bit Power and 8 Bit Compatibility
- Runs in Existing S-50 Systems where Physical Space Allows
- Full 24 Bit Addressing
- Fully Transparent Refresh

THE PRICES

Because of the variety of configurations possible, full pricing cannot be given. Representative prices are:

- 64K 6809 HELIX $1995
- 64K 68000 HELIX $2595
- 512K 6809 HELIX $4450
- 512K 68000 HELIX $4995

HAZELWOOD COMPUTER SYSTEMS

907 E. Terra, O'Fallon, Missouri 63366 (314) 281-1055

Dealer and OEM Inquiries Invited. We support our Dealers.

Optima is a Trademark of Scientific-Atlanta